Clustering
Lecture 9: Other Topics

Jing Gao
SUNY Buffalo

Outline

- **Basics**
  - Motivation, definition, evaluation
- **Methods**
  - Partitional
  - Hierarchical
  - Density-based
  - Mixture model
  - Spectral methods
- **Advanced topics**
  - Clustering ensemble
  - Clustering in MapReduce
  - Subspace clustering, co-clustering, semi-supervised clustering

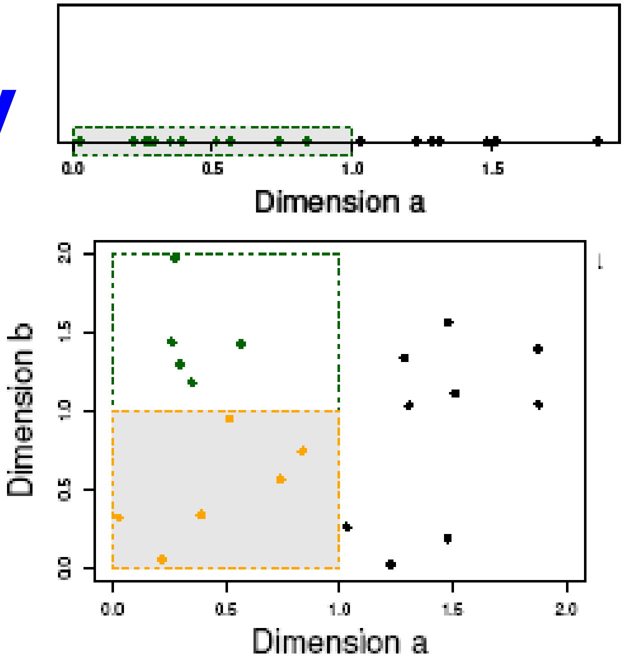
Clustering High-Dimensional Data

- **High-dimensional data everywhere**
  - Many applications: text documents, DNA micro-array data
  - Major challenges:
    - Many irrelevant dimensions may mask clusters
    - Distance measure becomes meaningless—due to equi-distance
    - Clusters may exist only in some subspaces

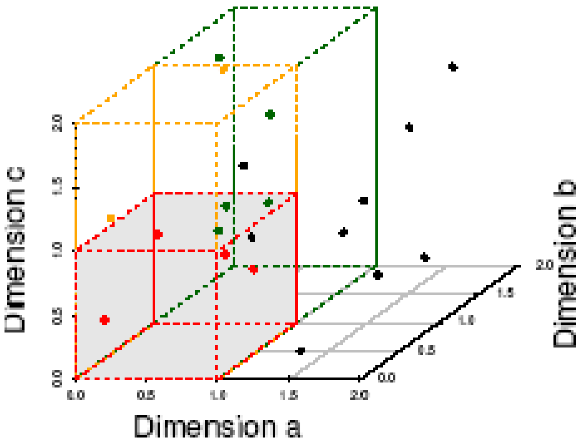
The Curse of Dimensionality

(graphs adapted from Parsons et al. KDD Explorations 2004)

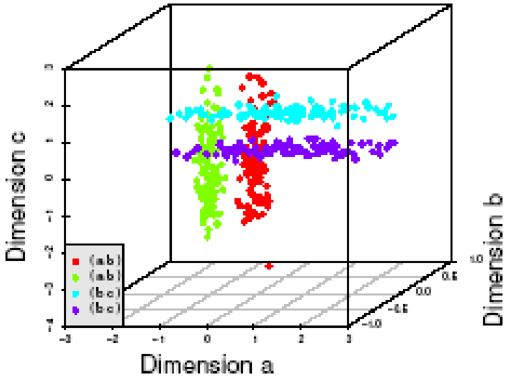
- Data in only one dimension is relatively packed
- Adding a dimension “stretch” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance



(b) 6 Objects in One Unit Bin



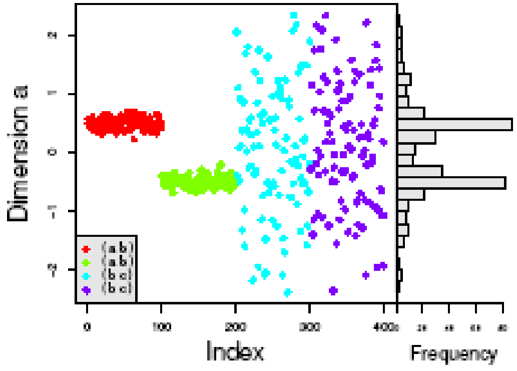
(c) 4 Objects in One Unit Bin



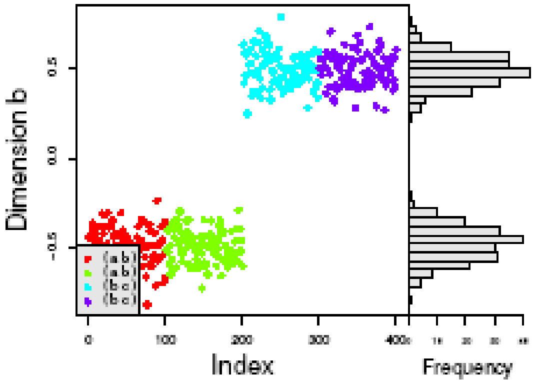
Why Subspace Clustering?

(adapted from Parsons et al. SIGKDD Explorations 2004)

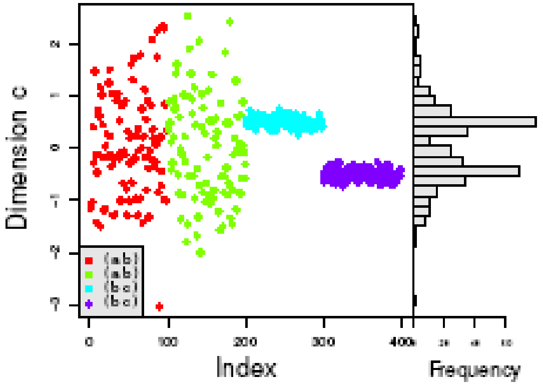
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



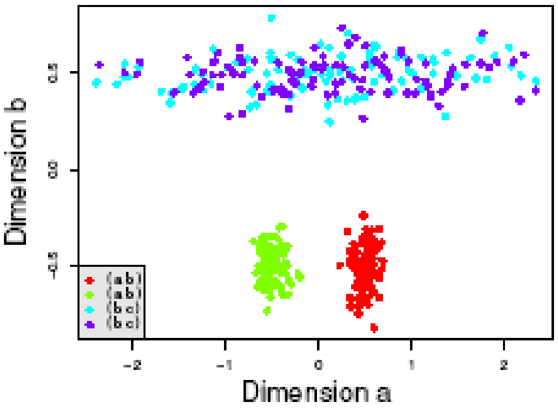
(a) Dimension *a*



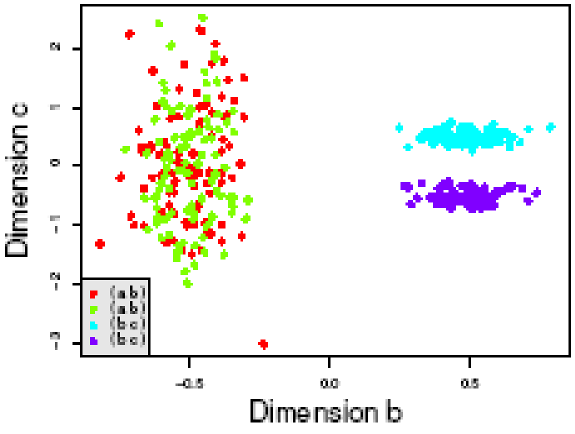
(b) Dimension *b*



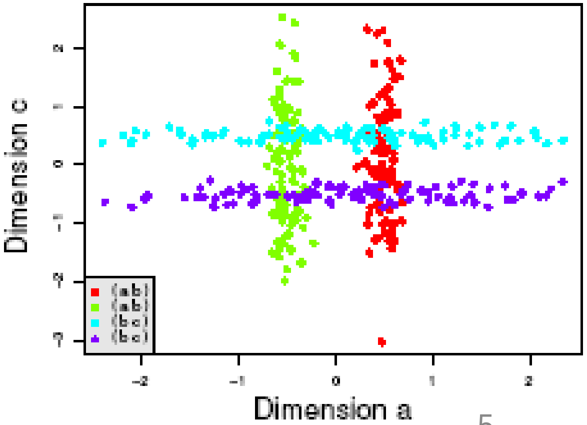
(c) Dimension *c*



(a) Dims *a* & *b*



(b) Dims *b* & *c*



(c) Dims *a* & *c*

CLIQUE (Clustering In QUEst)

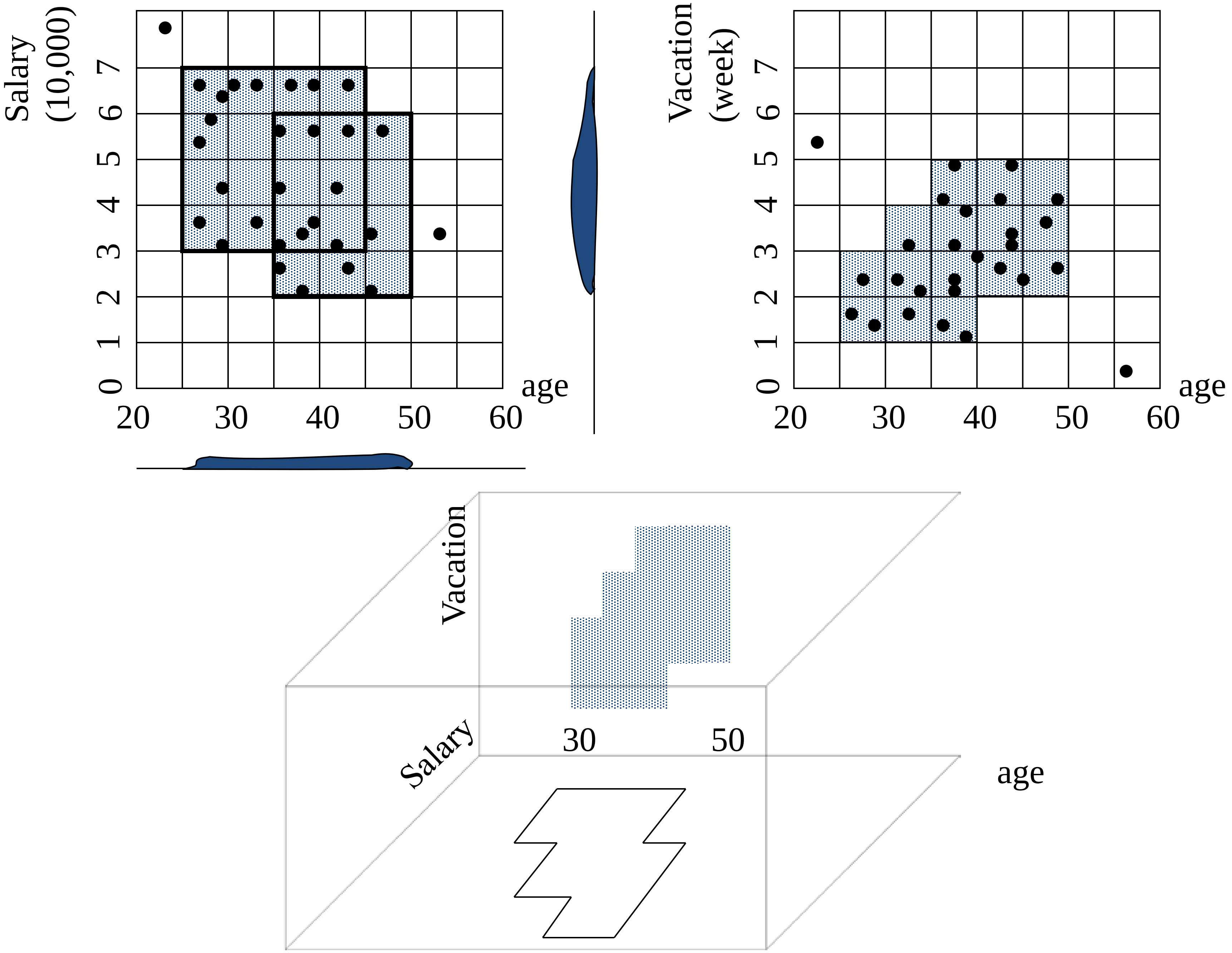
- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD’98)
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- **Basic idea of CLIQUE**
  - It partitions each dimension into the same number of equal length interval
  - It partitions an high dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps (1)

- **Grid density**
  - Partition the data space and find the number of points that lie inside each cell of the partition
- **Dense subspace**
  - Identify the subspaces that contain clusters using the Apriori principle
  - Dense subspace in (*d*+1)-dimension should be dense in *d*-dimension
  - Start with 1-d units and find the dense units in all the subspaces

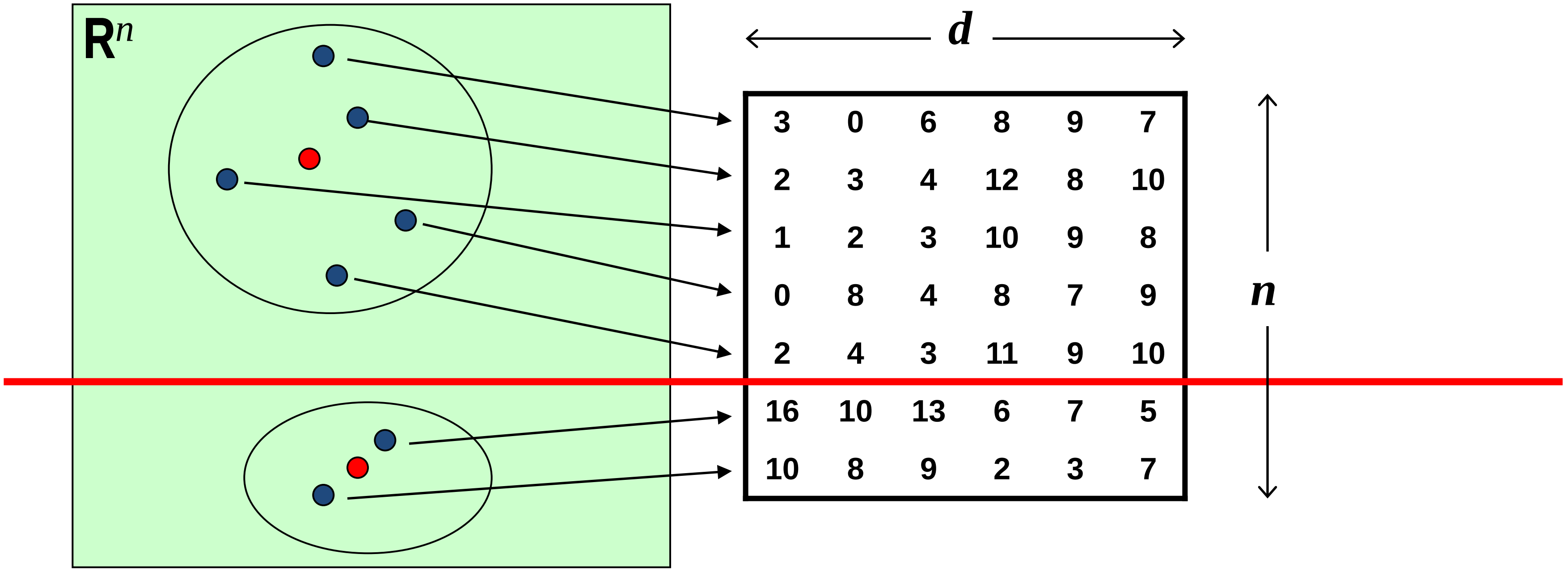
CLIQUE: The Major Steps (2)

- **Identify clusters**
  - Determine dense units in all subspaces
  - Determine connected dense units in all subspaces
- **Generate minimal description for the clusters**
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster


Salary (10,000)
0 1 2 3 4 5 6 7
20 30 40 50 60
age
Vacation (week)
0 1 2 3 4 5 6 7
20 30 40 50 60
age
Vacation
Salary
30
50
age

Clustering Definition Revisited

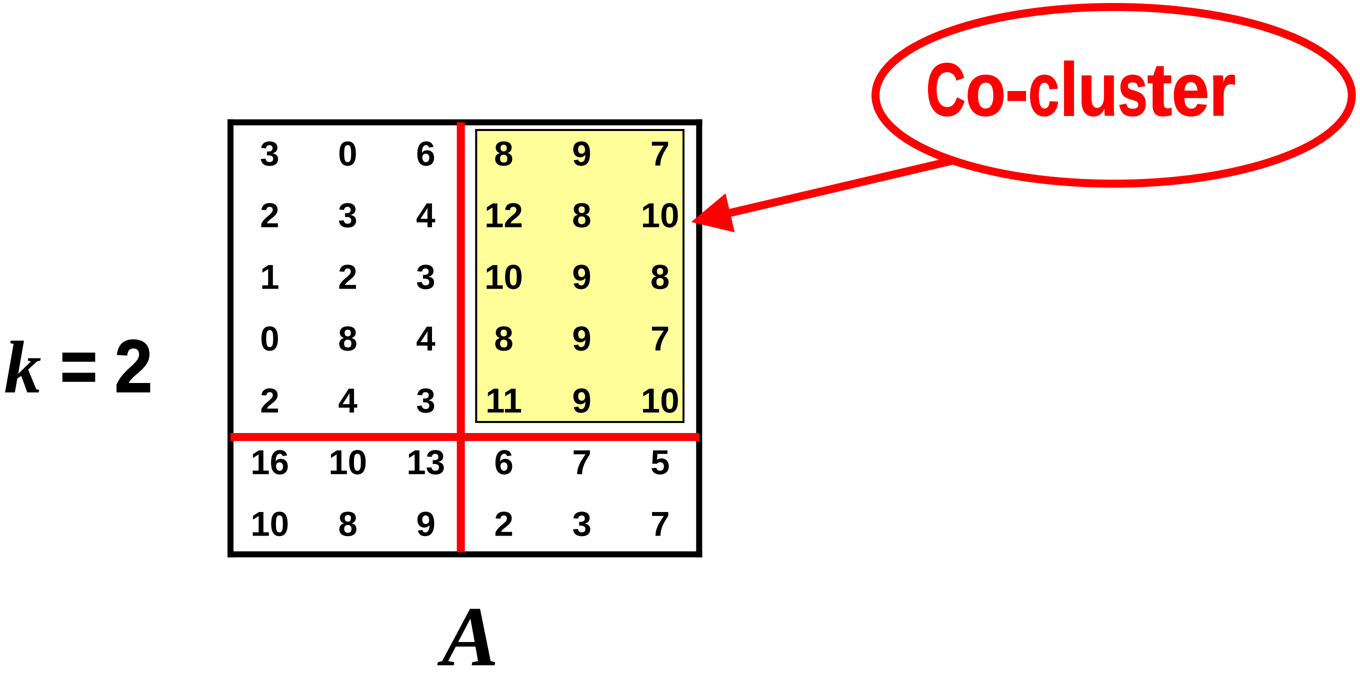
- *n* points in $\mathbf{R}^d$
- Group them to *k* clusters
- Represent them by a matrix *A*Î $\mathbf{R}^{n\times d}$
  - – A point corresponds to a row of *A*
- **Clustering:** Partition the rows to *k* clusters



3	0	6	8	9	7
2	3	4	12	8	10
1	2	3	10	9	8
0	8	4	8	7	9
2	4	3	11	9	10
16	10	13	6	7	5
10	8	9	2	3	7

Co-Clustering

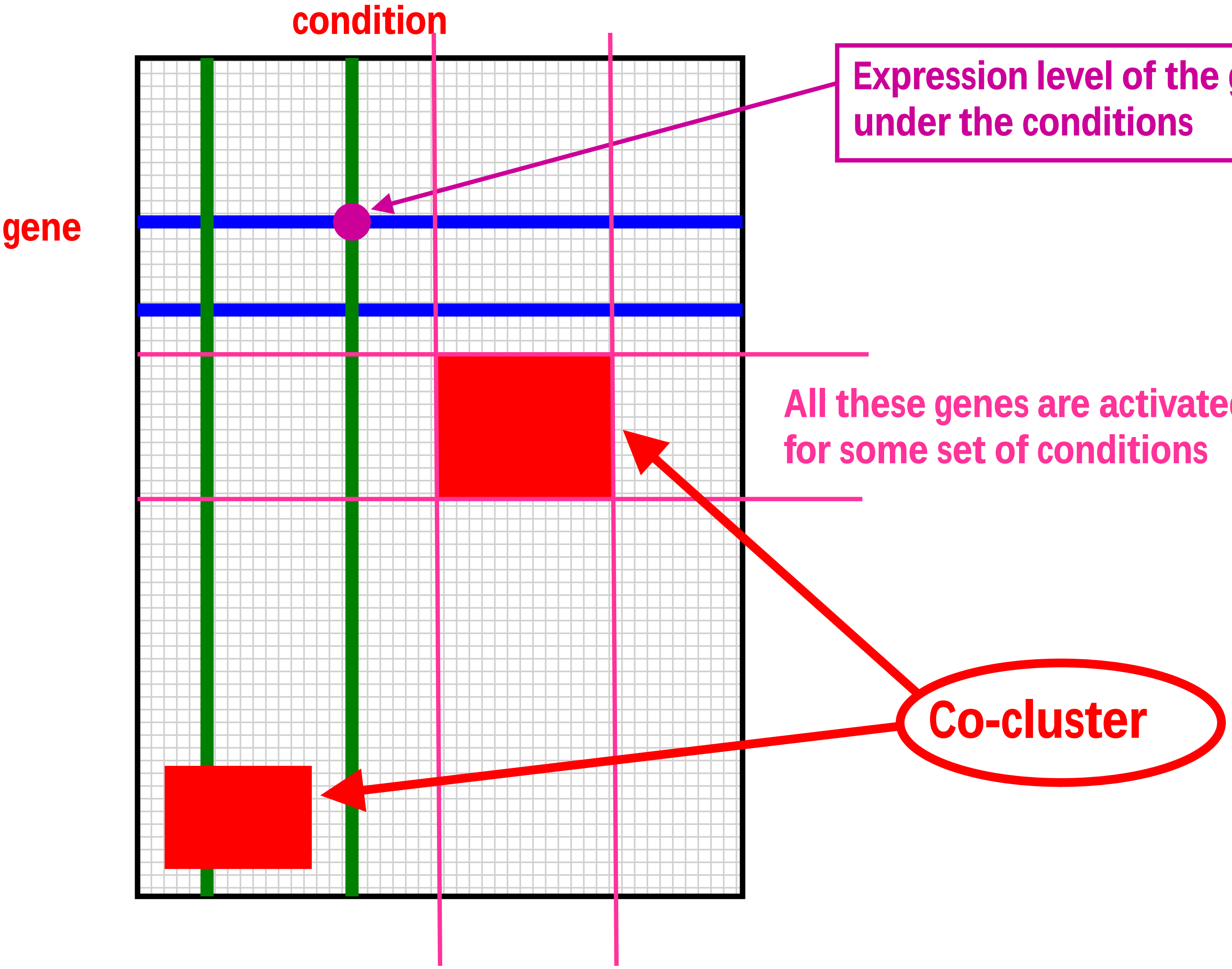
- **Co-Clustering**
  - Cluster rows and columns of A simultaneously:



$k = 2$

$$A = \left[\begin{array}{ccc|ccc} 3 & 0 & 6 & 8 & 9 & 7 \\ 2 & 3 & 4 & 12 & 8 & 10 \\ 1 & 2 & 3 & 10 & 9 & 8 \\ 0 & 8 & 4 & 8 & 9 & 7 \\ 2 & 4 & 3 & 11 & 9 & 10 \\ \hline 16 & 10 & 13 & 6 & 7 & 5 \\ 10 & 8 & 9 & 2 & 3 & 7 \end{array}\right]$$

Co-Clusters in Gene Expression Data



Expression level of the gene under the conditions

All these genes are activated for some set of conditions

K-Means Objective Function Revisited

3	0	6	8	9	7
2	3	4	12	8	10
1	2	3	10	9	8
0	8	4	8	7	9
2	4	3	11	9	10
16	10	13	6	7	5
10	8	9	2	3	7

Original data points **A**

1.6	3.4	4	9.8	8.4	8.8
1.6	3.4	4	9.8	8.4	8.8
1.6	3.4	4	9.8	8.4	8.8
1.6	3.4	4	9.8	8.4	8.8
1.6	3.4	4	9.8	8.4	8.8
13	9	11	4	5	6
13	9	11	4	5	6

Data representation **A’**

- In **A’,** every point in **A** is replaced by the corresponding cluster center
- The quality of the clustering is measured by computing distances between the data entries of **A** and **A’**

$$\min \sum_{j} \sum_{x \in C_k} (x - m_k)^2 \quad \Rightarrow \quad \min \sum_{i} \sum_{j} (A_{ij} - A'_{ij})^2$$

Co-Clustering Objective Function

3	0	6	8	9	7
2	3	4	12	8	10
1	2	3	10	9	8
0	8	4	8	7	9
2	4	3	11	9	10
16	10	13	6	7	5
10	8	9	2	3	7

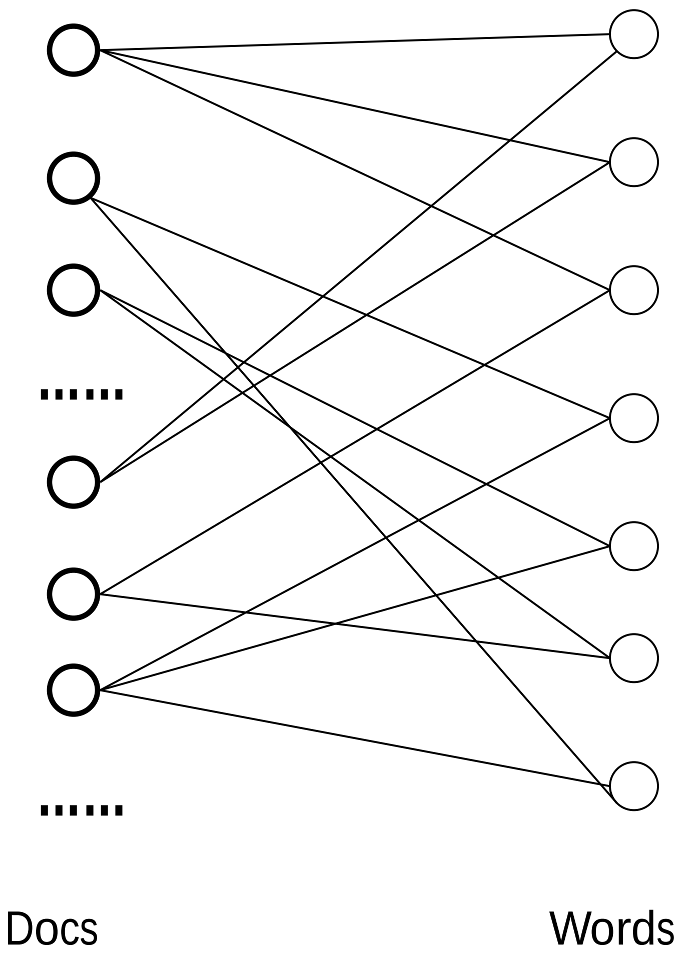
3	3	3	9	9	9
3	3	3	9	9	9
3	3	3	9	9	9
3	3	3	9	9	9
3	3	3	9	9	9
11	11	11	5	5	5
11	11	11	5	5	5

- In **A'** every point in **A** is replaced by the corresponding co-cluster center
- The quality of the clustering is measured by computing distances between the data in the cells of **A** and **A'**

$$\min \sum_{i,j} \sum_{x_{ij} \in C_k} (x_{ij} - m_k)^2 \quad \Longrightarrow \quad \min \sum_{i} \sum_{j} (A_{ij} - A'_{ij})^2$$

Co-Clustering by Bipartite Graph Partitioning

- **Example**
  - Find co-clusters in documents
  - Co-clusters indicate that a set of keywords frequently occur together in a set of documents
- **Bipartite graph formulation**
  - Document-word association
- **Bipartite graph partitioning**
  - Result partitions are co-clusters



Probabilistic Models for Co-Clustering

- **Mixture model for clustering**
  - first pick one of the components with probability π_k
  - then draw a sample x_i from that component distribution
- **Co-clustering**
  - first pick one of the row clusters with probability p_r
  - first pick one of the column clusters with probability p_c
  - then draw a sample x_i from the co-cluster distribution (combination of row and column clusters forms a co-cluster)

Semi-supervised Clustering: Problem Definition

- Input:
  - A set of unlabeled objects, each described by a set of attributes
  - A small amount of domain knowledge
- Output:
  - A partitioning of the objects into *k* clusters
- Objective:
  - Maximum intra-cluster similarity
  - Minimum inter-cluster similarity
  - High consistency between the partitioning and the domain knowledge

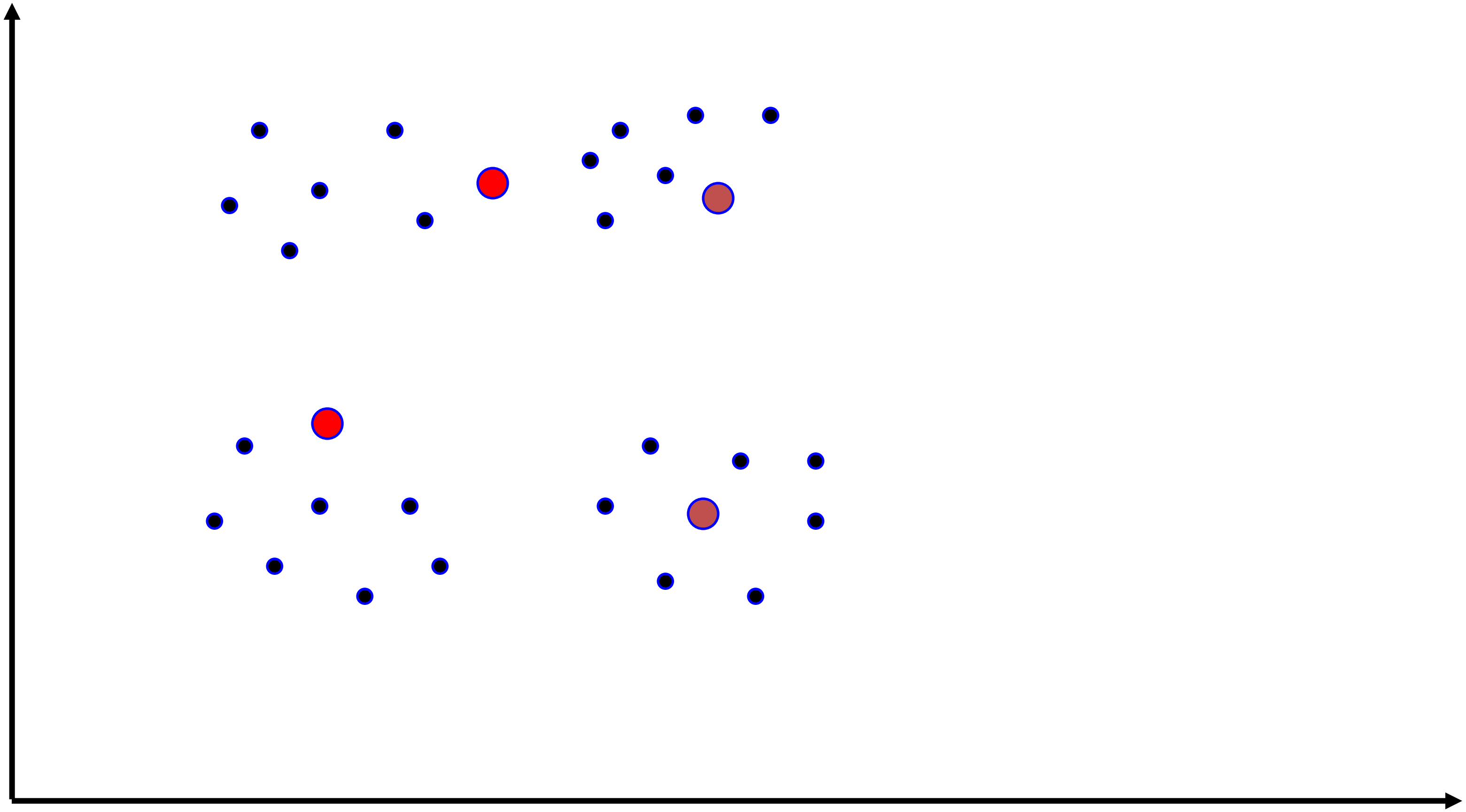
Semi-Supervised Clustering

- **Domain knowledge**
  - Partial label information is given
  - Apply some constraints (must-links and cannot-links)
- **Approaches**
  - Search-based Semi-Supervised Clustering
    - Alter the clustering algorithm using the constraints
  - Similarity-based Semi-Supervised Clustering
    - Alter the similarity measure based on the constraints
  - Combination of both

Semi-Supervised K-Means for partially labeled data

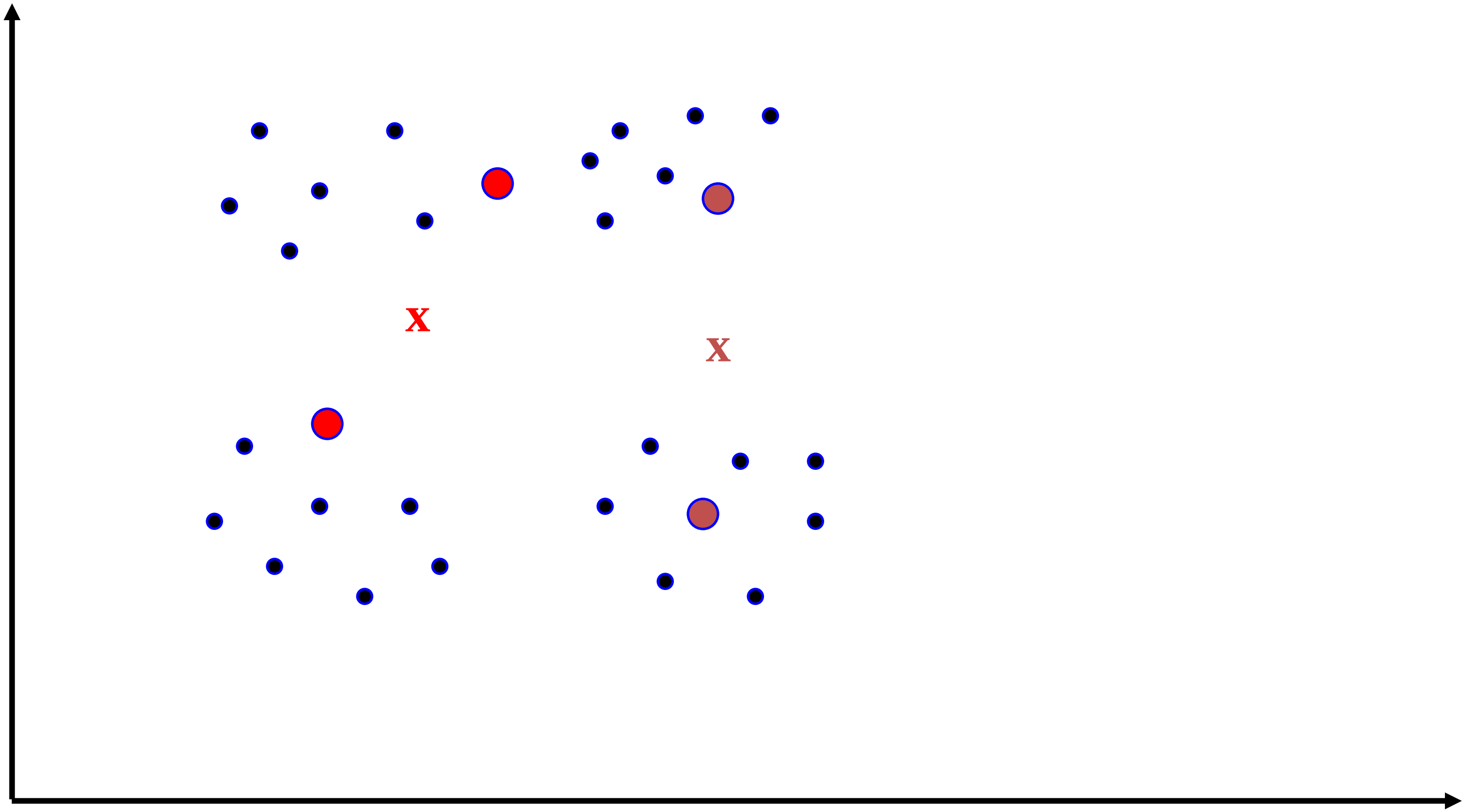
- **Seeded K-Means:**
  - Labeled data provided by user are used for initialization: initial center for cluster *i* is the mean of the seed points having label *i.*
  - Seed points are only used for initialization, and not in subsequent steps.
- **Constrained K-Means:**
  - Labeled data provided by user are used to initialize K-Means algorithm.
  - Cluster labels of seed data are kept unchanged in the cluster assignment steps, and only the labels of the non-seed data are re-estimated.

Seeded K-Means Example



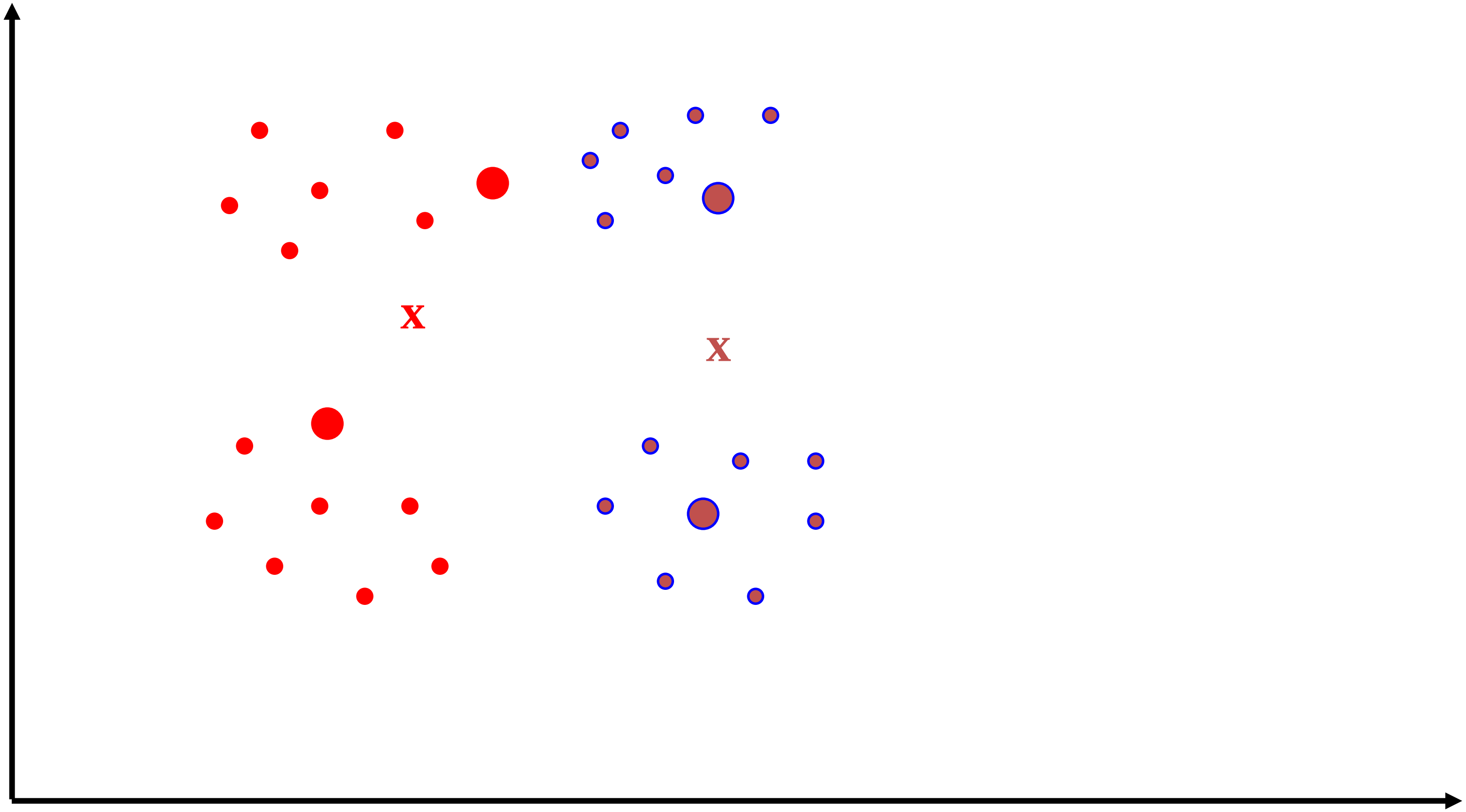
Seeded K-Means Example

Initialize Means Using Labeled Data



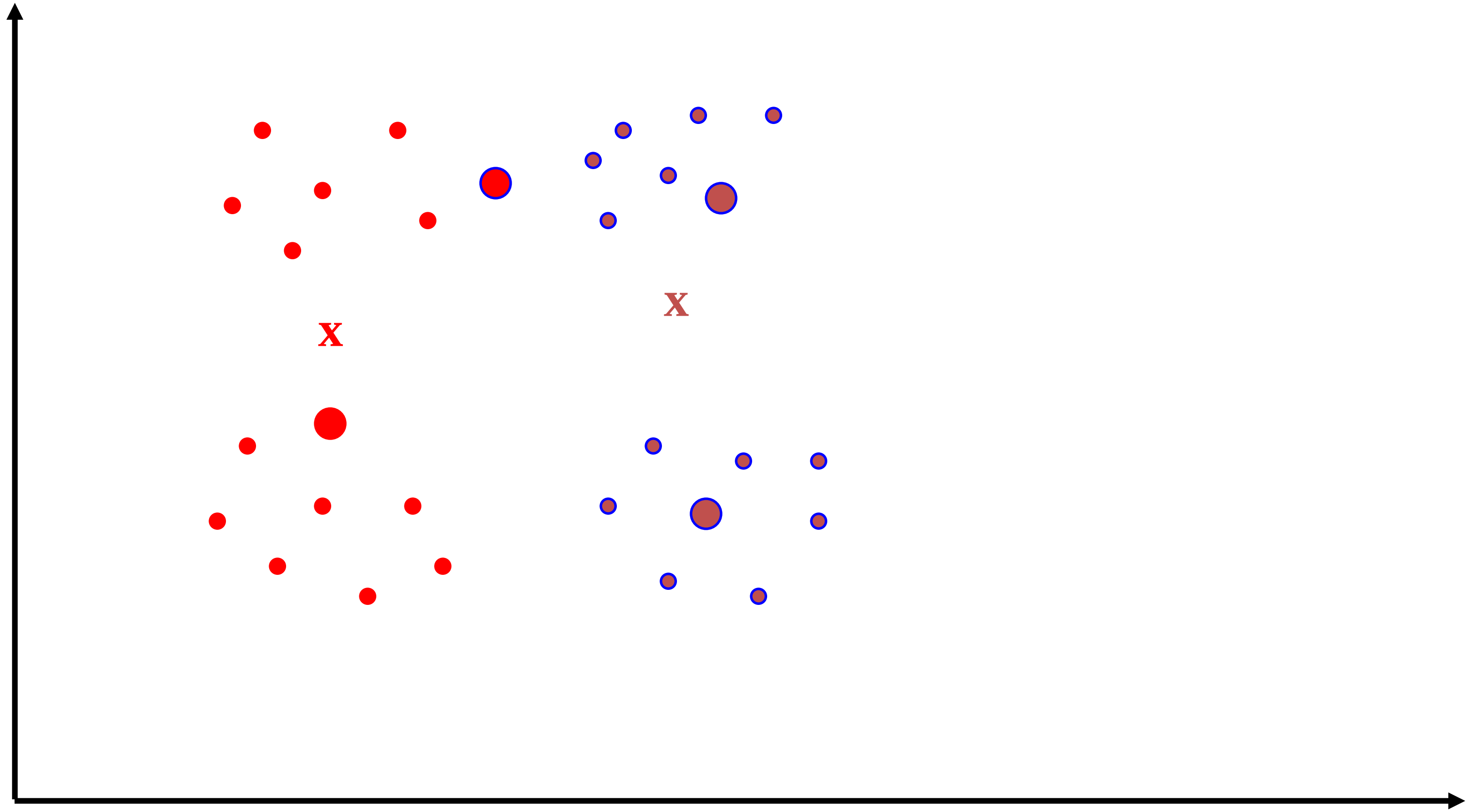
Seeded K-Means Example

Assign Points to Clusters



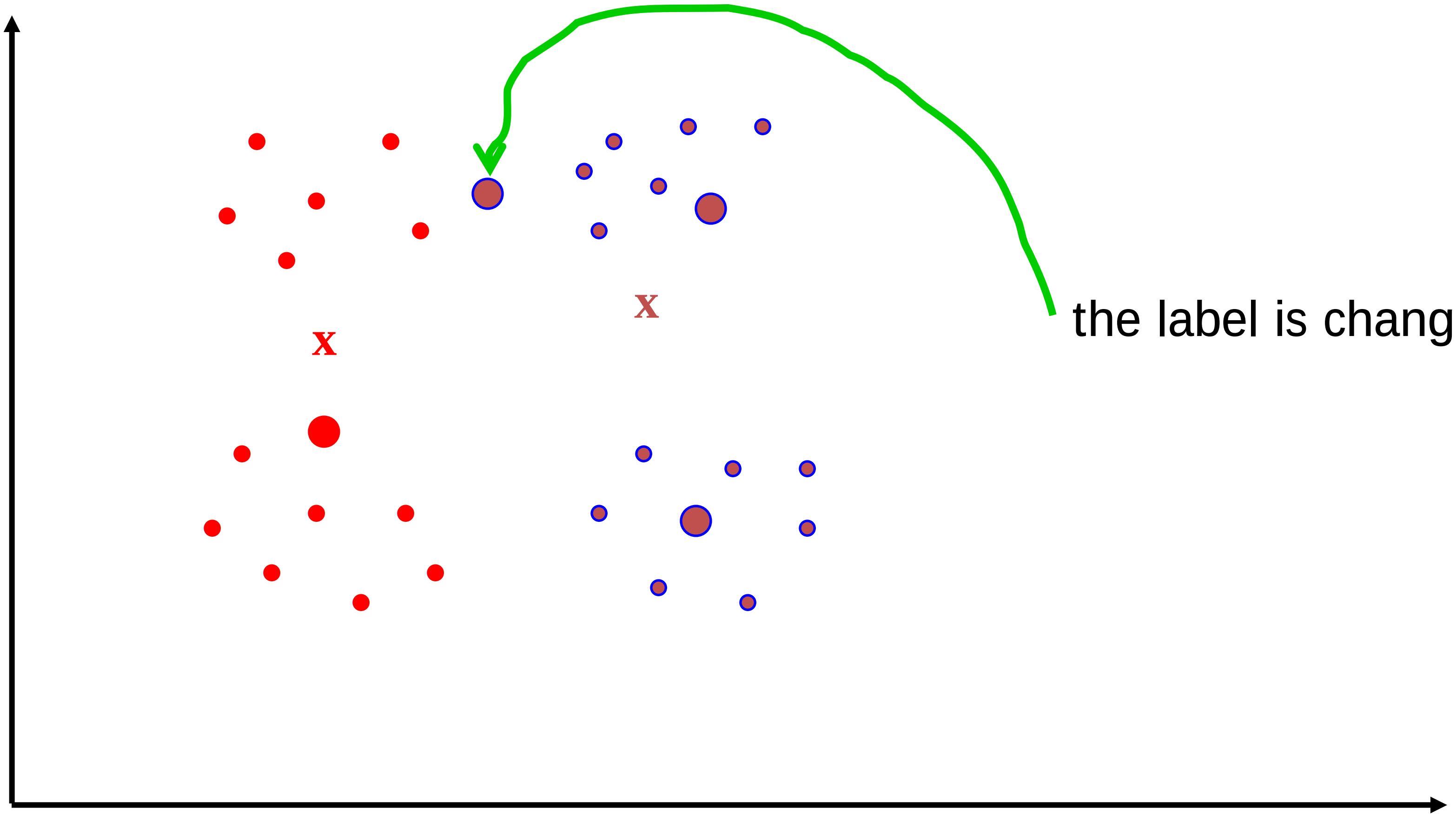
Seeded K-Means Example

Re-estimate Means



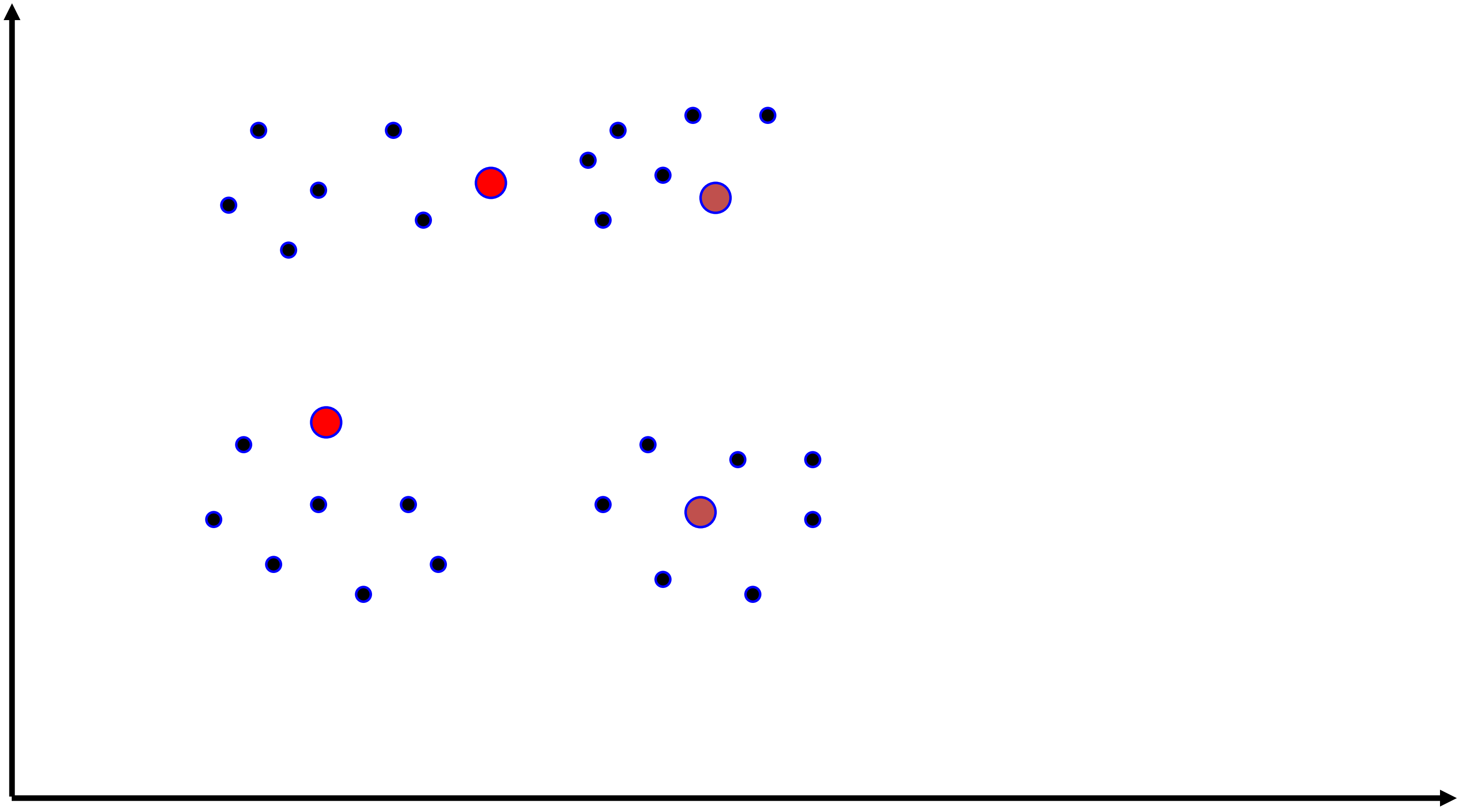
Seeded K-Means Example

Assign points to clusters and Converge



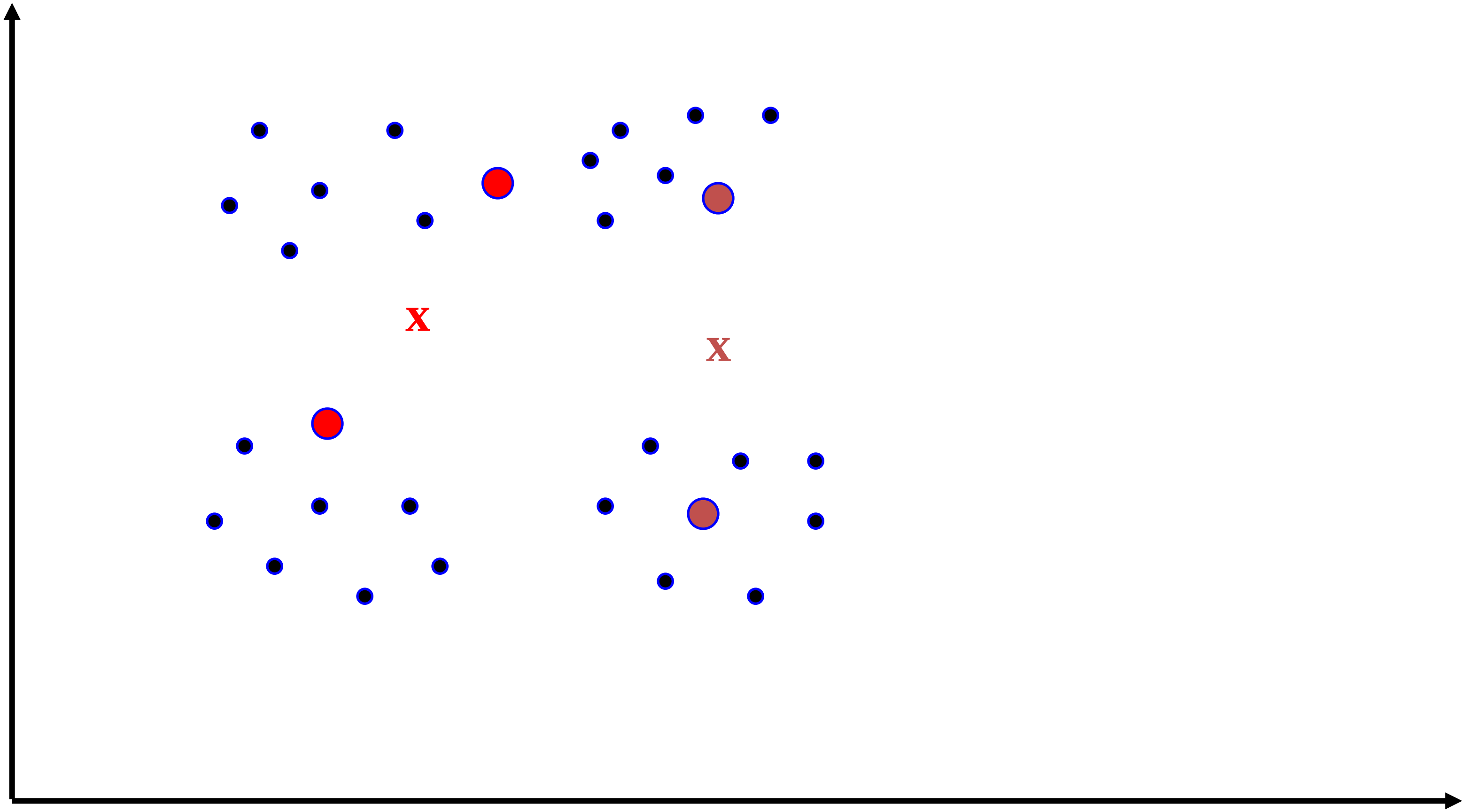
the label is changed

Constrained K-Means Example



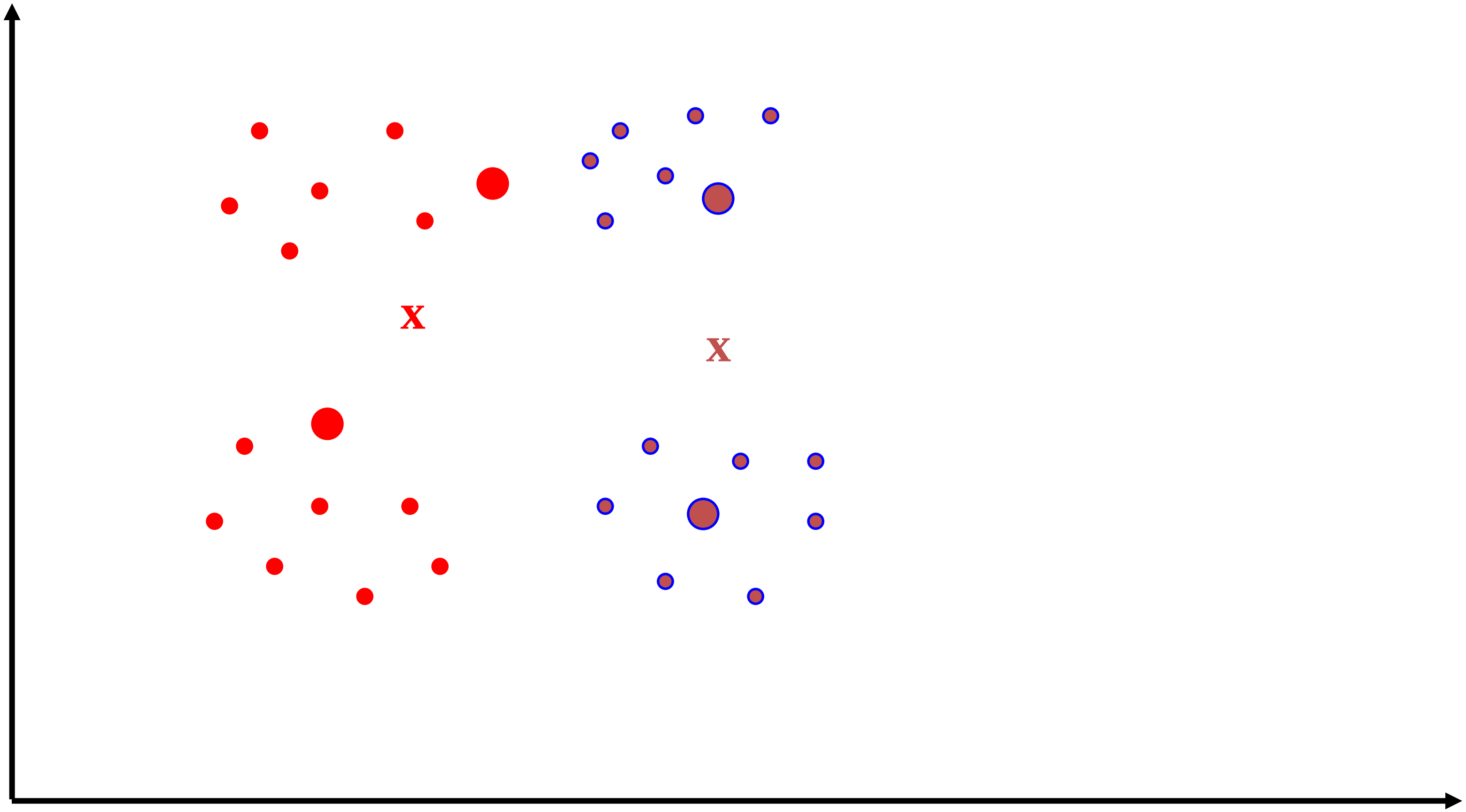
Constrained K-Means Example

Initialize Means Using Labeled Data



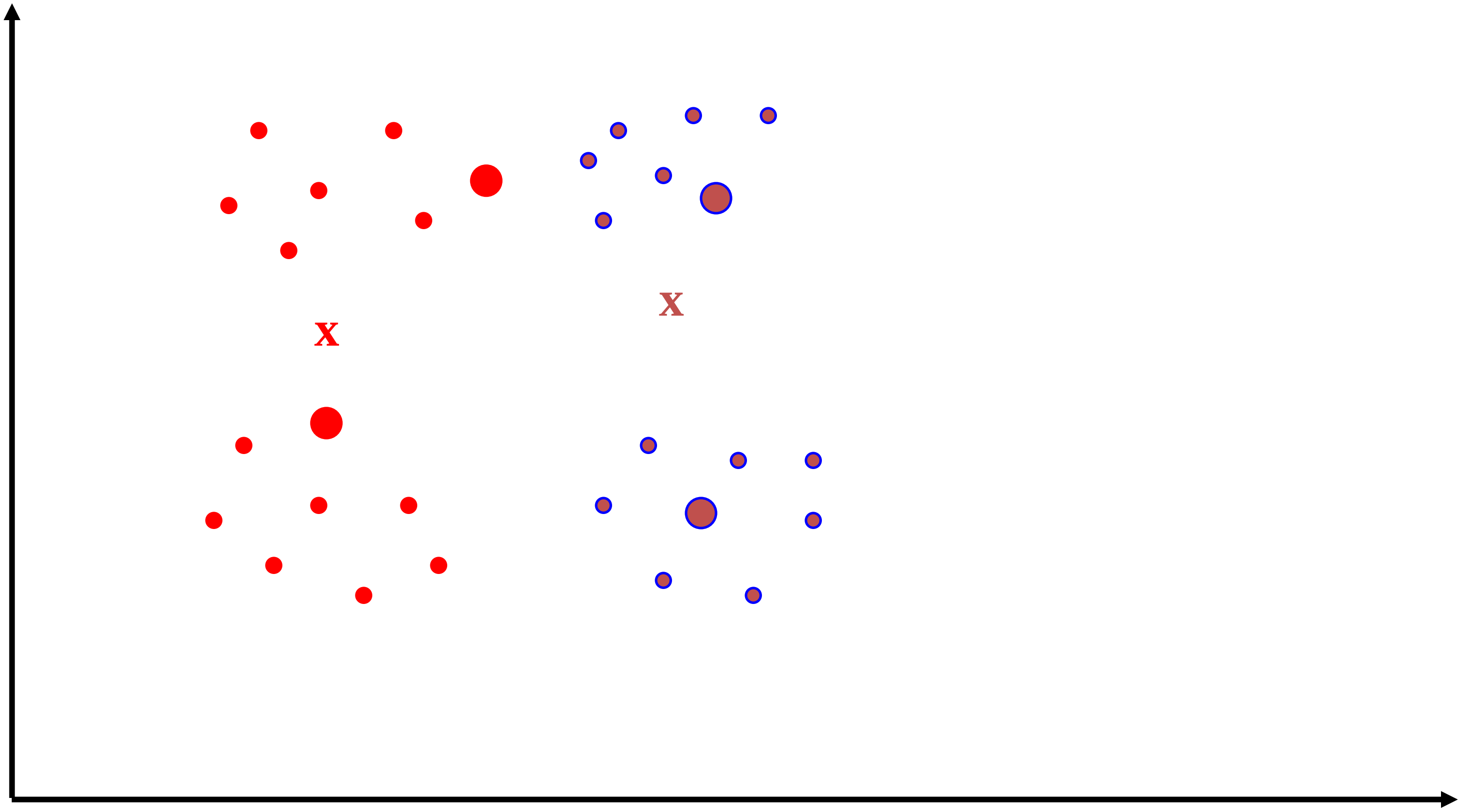
Constrained K-Means Example

Assign Points to Clusters



Constrained K-Means Example

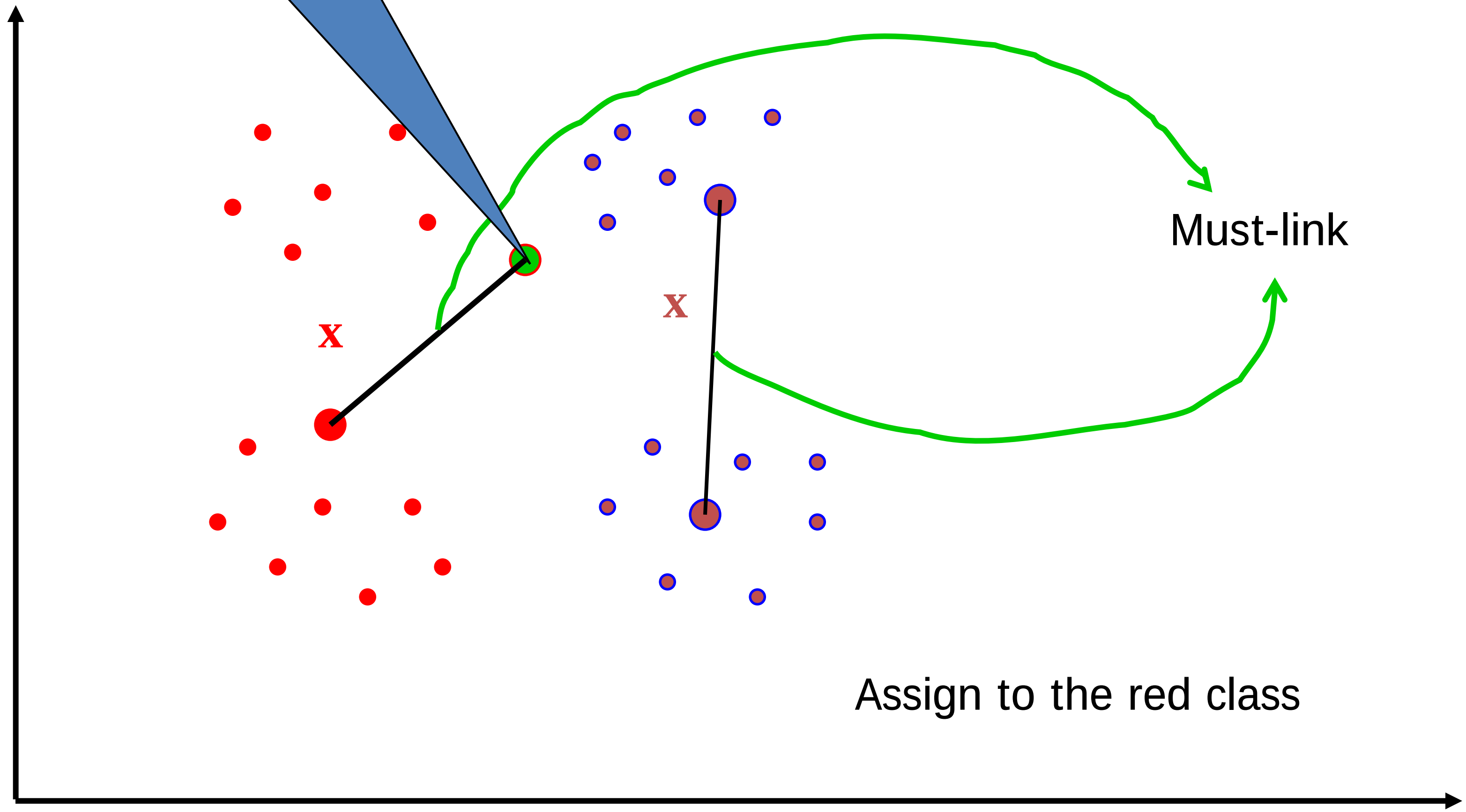
Re-estimate Means and Converge



COP K-Means

- COP K-Means [Wagstaff *et al*.: ICML01] is K-Means with must-link (must be in same cluster) and cannot-link (cannot be in same cluster) constraints on data points.
- **Initialization**
  - Cluster centers are chosen randomly
- **Algorithm**
  - During cluster assignment step in COP-K-Means, a point is assigned to its nearest cluster without violating any of its constraints. If no such assignment exists, abort.

COP K-Means Algorithm

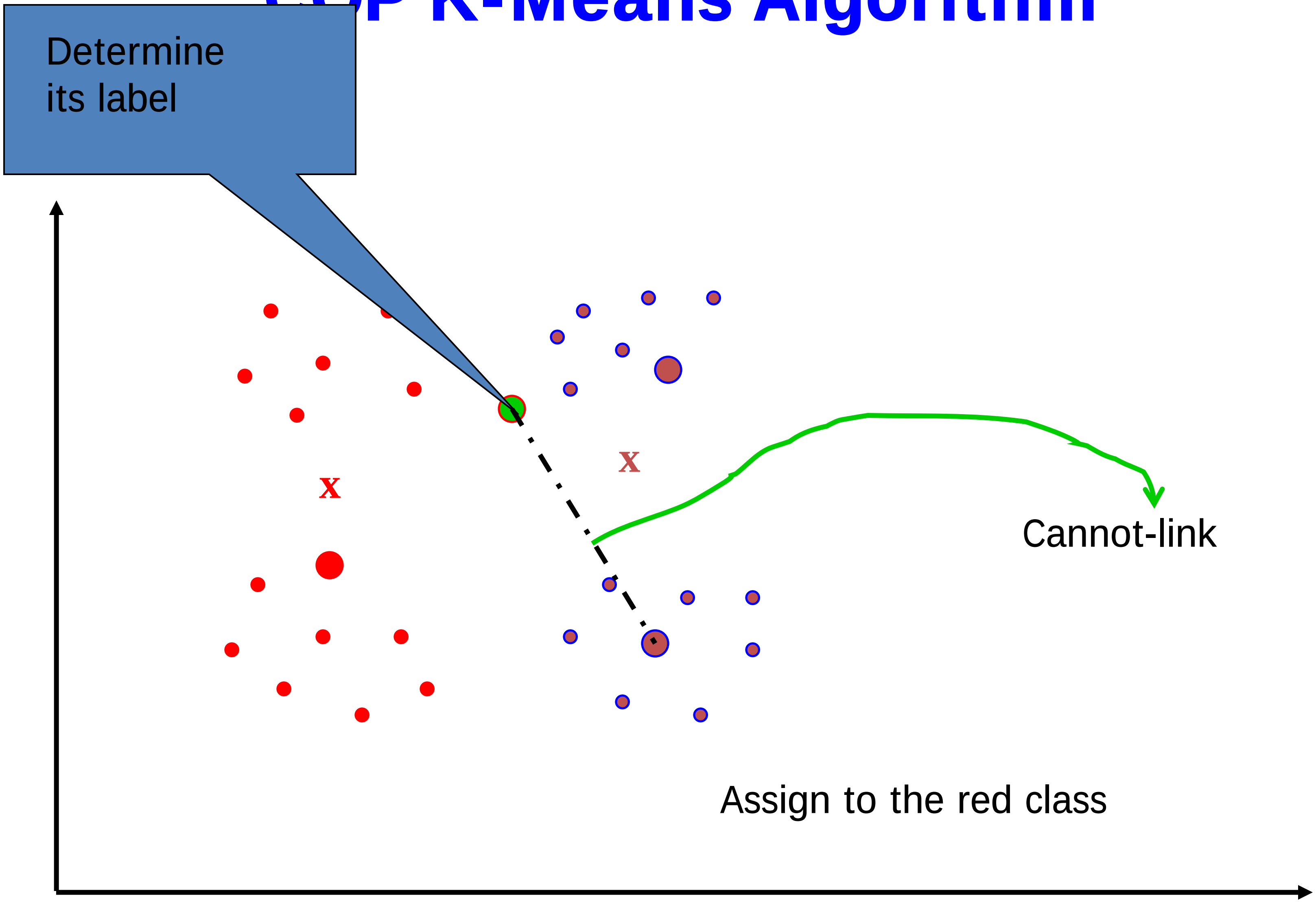


Determine its label

Must-link

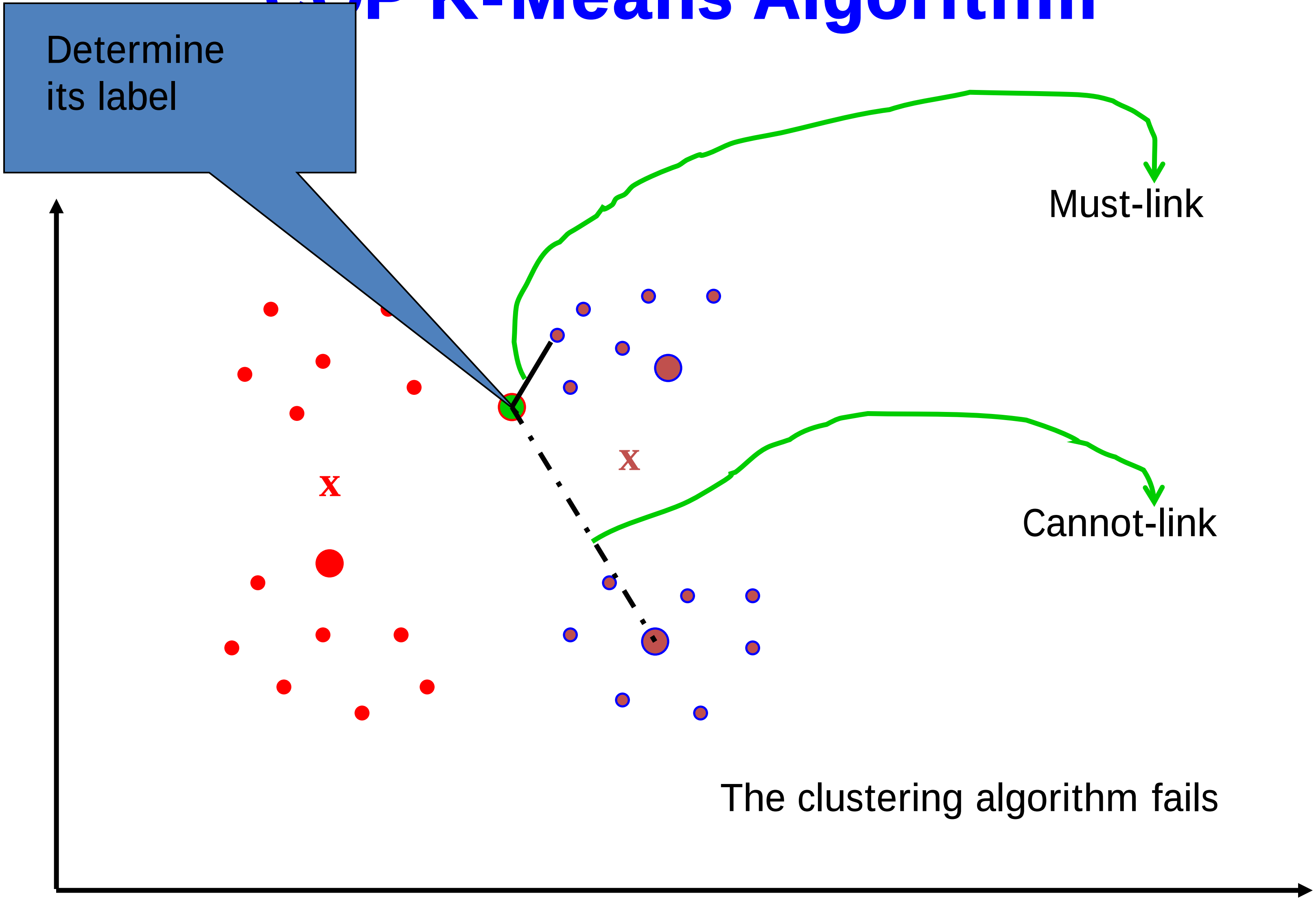
Assign to the red class

COP K-Means Algorithm



Assign to the red class

COP K-Means Algorithm



Determine its label

Must-link

Cannot-link

The clustering algorithm fails

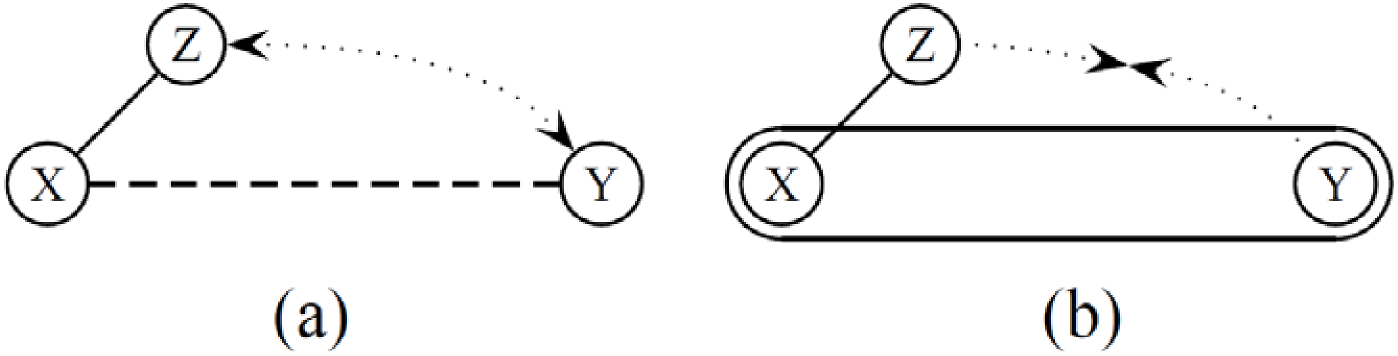
Similarity-Based Semi-Supervised Clustering

- Train an adaptive similarity function to fit the labeled data
- Use a standard clustering algorithm with the trained similarity function to cluster the unlabeled data
- Adaptive similarity functions:
  - Altered similarity matrix [Kamvar:IJCAI03]
  - Trained Mahalanobis distance [Xing:NIPS02]
  - Altered Euclidian distance [Klein:ICML02]
- Clustering algorithms:
  - Spectral clustering [Kamvar:IJCAI03]
  - Complete-link agglomerative [Klein:ICML02]
  - K-means [Xing:NIPS02]

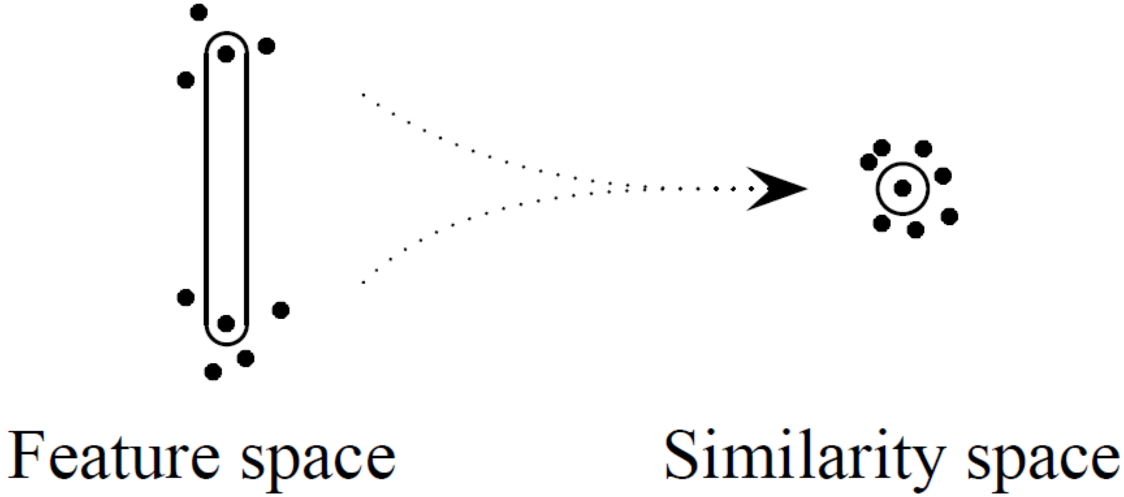
Using Constraints to Alter Similarity

Cannot Link—Pull points away

Must Link—Drag points close



Must Link—Drag points close



Altered similarity matrix

- Paper: Spectral learning. Kamvar *et al.*

- Graph based clustering
  - W: similarity matrix
  - D: degree matrix (row sum of W)

- Key idea: alter the similarity matrix W based on the domain knowledge

Semi-supervised spectral clustering

1. Compute the similarity matrix W and D
2. For each pair of must-link (i,j), assign $W_{ij} = W_{ji} = 1$
3. For each pair of cannot-link (i,j), assign $W_{ij} = W_{ji} = 0$
4. Form the matrix $D^{-0.5} W D^{-0.5}$
5. Form the matrix Y consisting of the first K eigenvectors of $D^{-0.5} W D^{-0.5}$
6. Normalize Y so that all the rows have unit lengths
7. Run K-Means on the rows to get the K clusters

Distance metric learning

Paper: Distance metric learning, with application to clustering with side-information. E. Xing, *et al.*

Given two sets of pairs S and D:

$$\mathrm{S}:\ (\mathrm{x}_i,\mathrm{x}_j)\in \mathrm{S}, \text{if } \mathrm{x}_i \text{ and } \mathrm{x}_j \text{ are similar}$$

$$\mathrm{D}:(\mathrm{x}_i,\mathrm{x}_j)\in \mathrm{D}, \text{if } \mathrm{x}_i \text{ and } \mathrm{x}_j \text{ are disimilar}$$

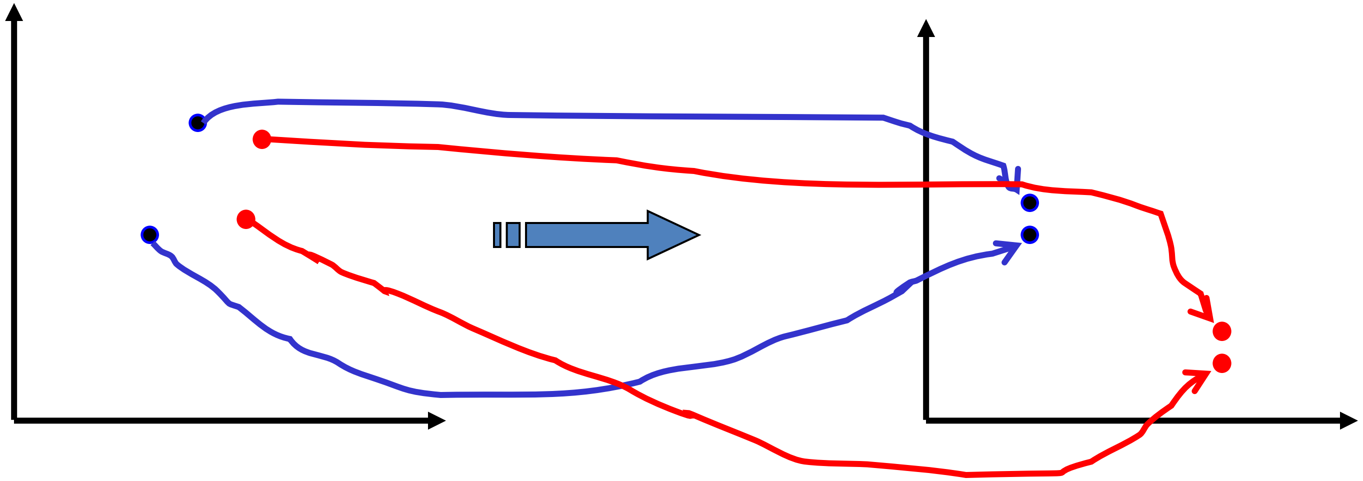
Compute a distance metric which respects these two sets

Distance metric learning

Define a new distance measure of the form:

$$\boxed{d(x,y) = \|x-y\|_A = \sqrt{(x-y)^T A (x-y)} \qquad A \geq 0}$$

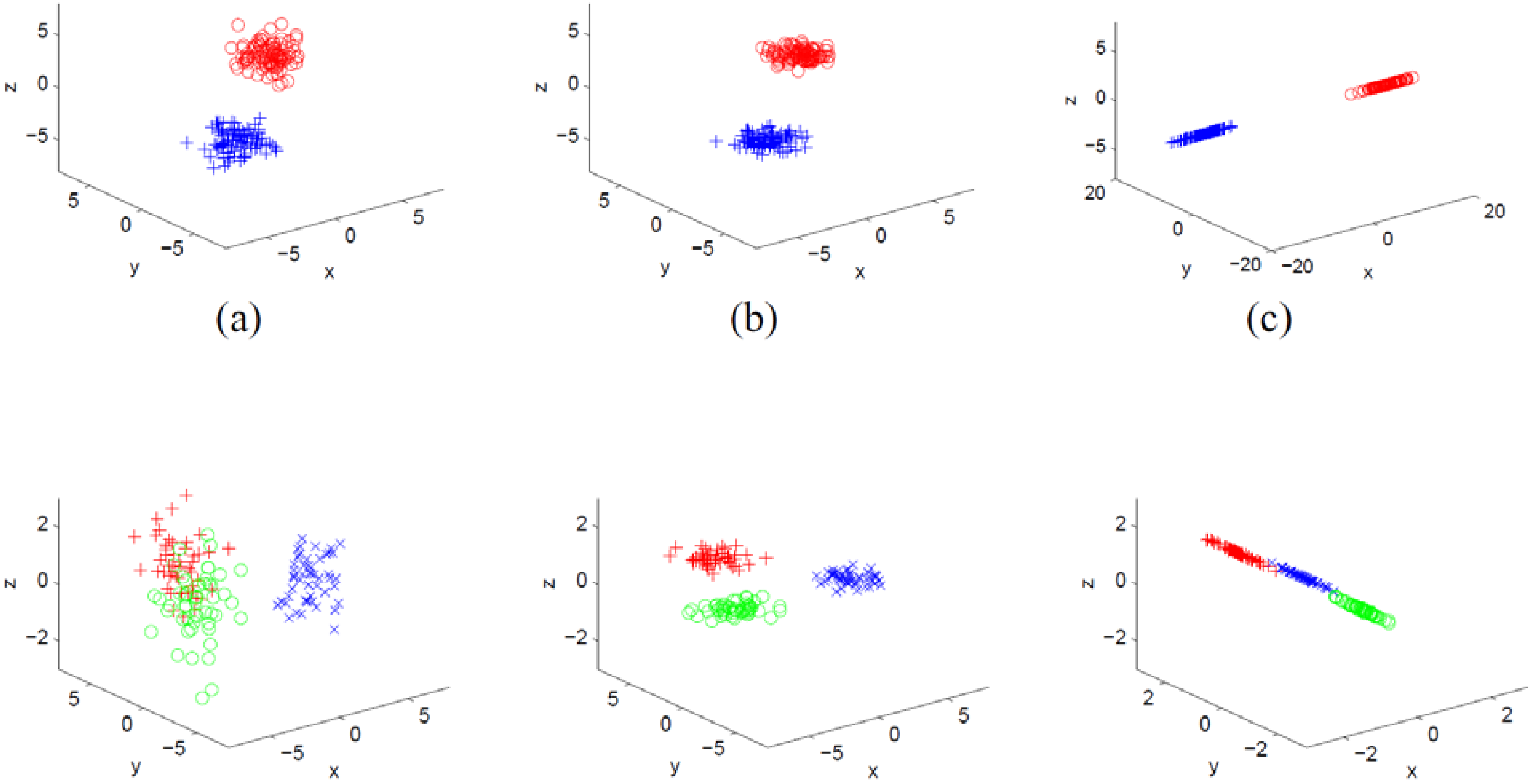
$x \to A^{1/2}x$ Linear transformation of the original data



Original Data

A-Diagonal Matrix
(Rescaling)

A-Full Matrix
(Any transformation)



(a) (b) (c)

(a) (b) (c)

Source: E. Xing, et al. Distance metric learning

Take-away Message

- Subspace clustering tries to find clusters in subspaces in high-dimensional data
- Co-clustering tries to find strong associations among a set of objects with respect to a set of attributes
- Semi-supervised clustering tries to improve clustering based on existing domain knowledge (labeled data or pairwise constraints)
- Many other topics to be explored for clustering ……