Constrained spectral embedding for K-way data clustering

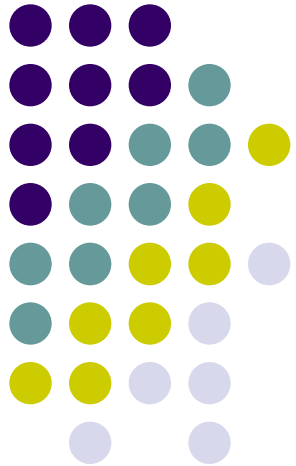
Denis Hamad

LISIC – ULCO

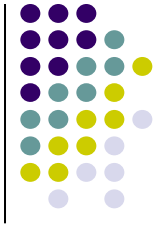
50 rue F. Buisson,

BP 719 62228 Calais Cedex

Denis.hamad@lisic.univ-littoral.fr



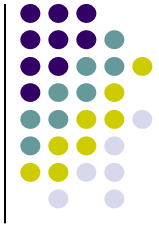
Outlines



Part 1:

- Data context
  - Data form: matrix rectangular
  - Data form: matrix square
- Supervised, unsupervised, semi-supervised contexts
- Feature selection - feature extraction
- Classification approaches
- Data clustering and cluster assumptions

Outlines



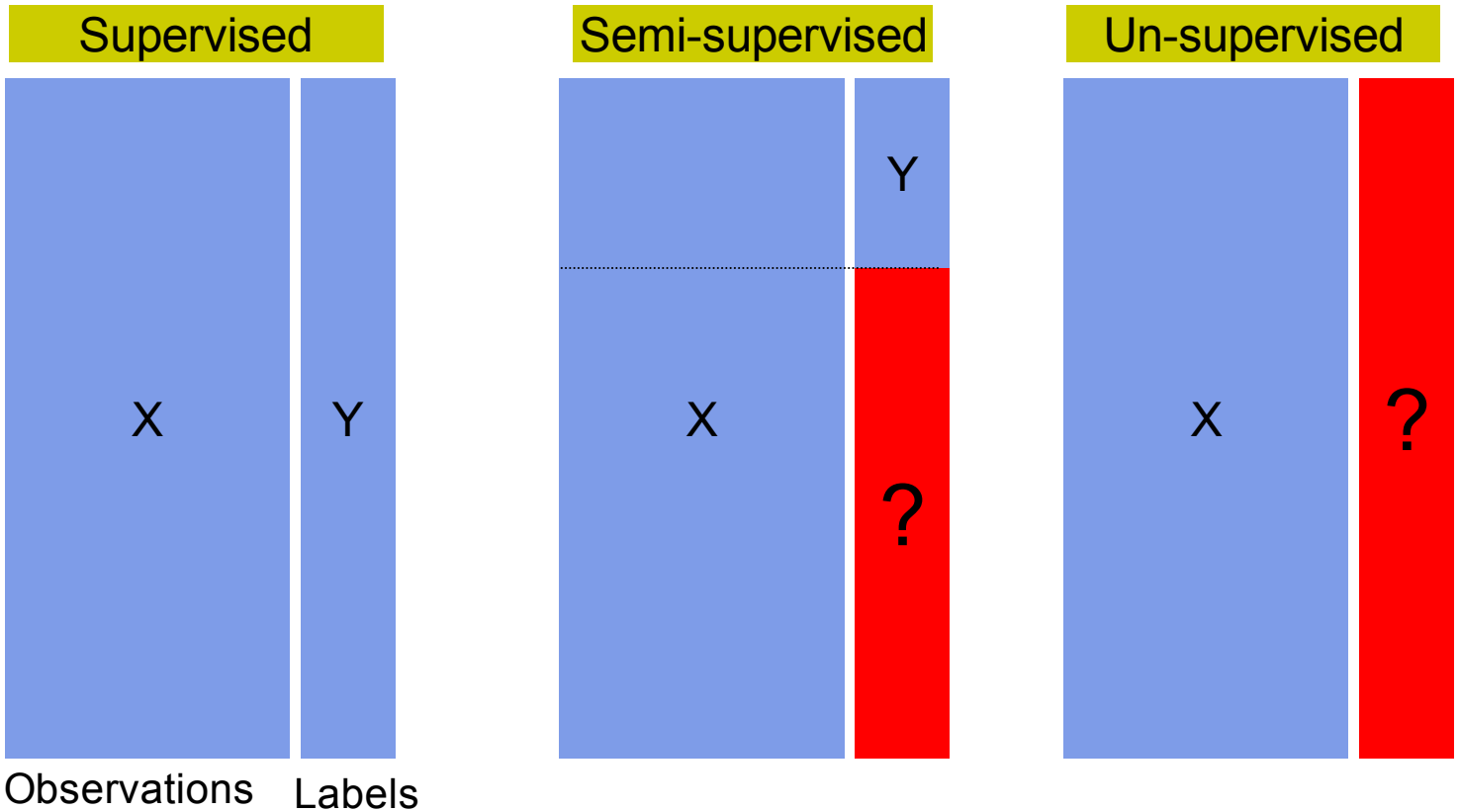
Part 2:

- Spectral clustering
  - Graph theory
  - Illustrative example
  - Spectral clustering algorithms
  - Parameters tuning

Part 3:

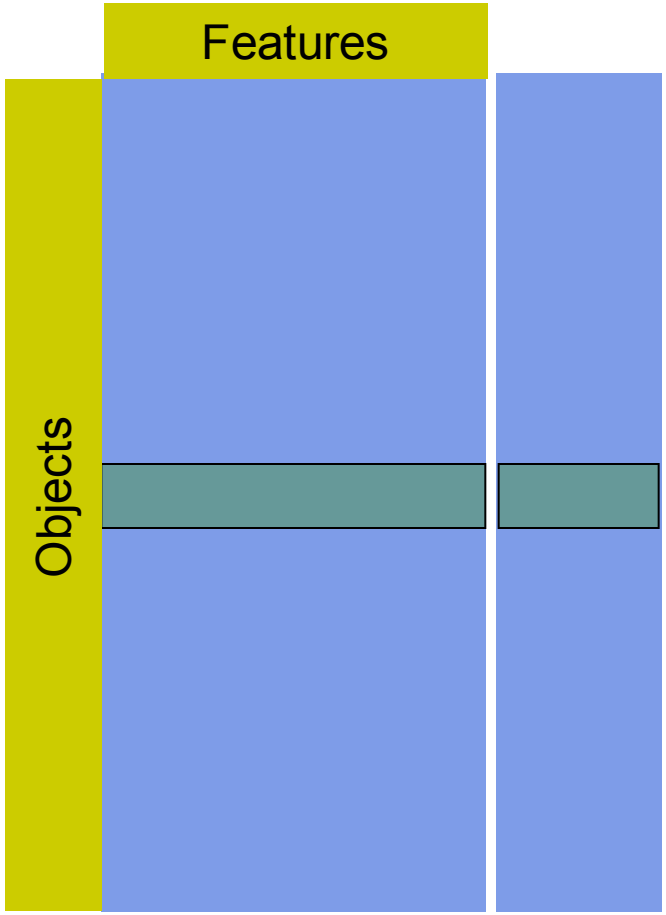
- Semi-supervised dimensionality reduction
  - Principal Component Analysis (PCA) and constrained PCA
  - Locality preserving projection (LLP) and constrained LPP
  - Spectral clustering and constrained spectral clustering

Data form in classification contexts



Data forms in unsupervised context

Data of vector form $\in R^l$



Obj / Obj	X_1	X_2	...	X_n
X_1	$s(x_1,x_1)$	$s(x_1,x_2)$		$s(x_1,x_n)$
X_2	$s(x_2,x_1)$	$s(x_2,x_2)$		$s(x_2,x_n)$
: .				
X_n	$s(x_n,x_1)$	$s(x_n,x_2)$	…	$s(x_n,x_n)$

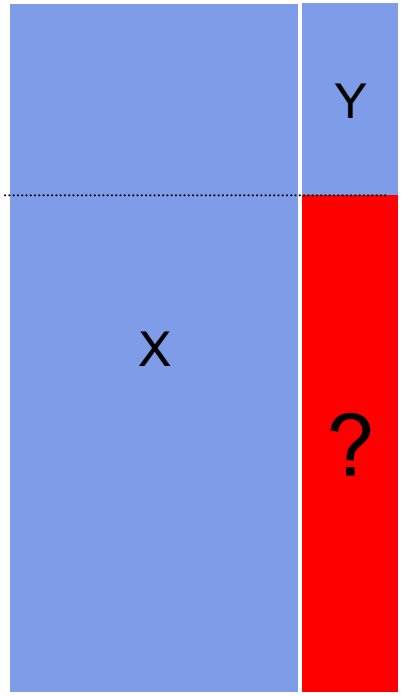
$$s_{ij} = f(d(X_i, X_j))$$

Data forms in semi-supervised context

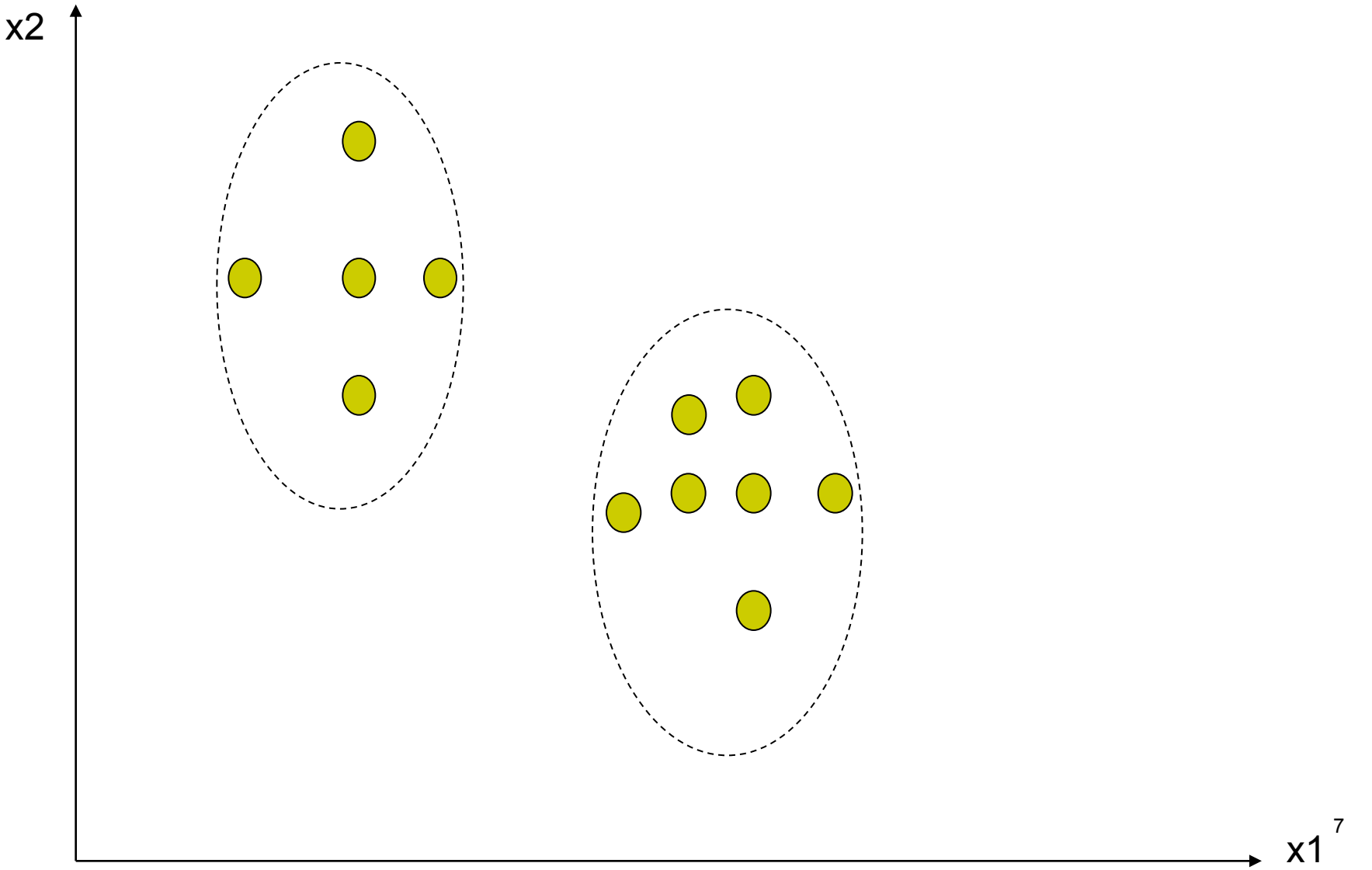
Data of similarity matrix form $\in R^n x R^n$

	X_1	X_2	...	X_n
X_1	$s(x_1,x_1)$	**0**		**1**
X_2	**0**	$s(x_2,x_2)$		$s(x_2,x_n)$
: .				
X_n	**1**	$s(x_n,x_2)$	...	$s(x_n,x_n)$

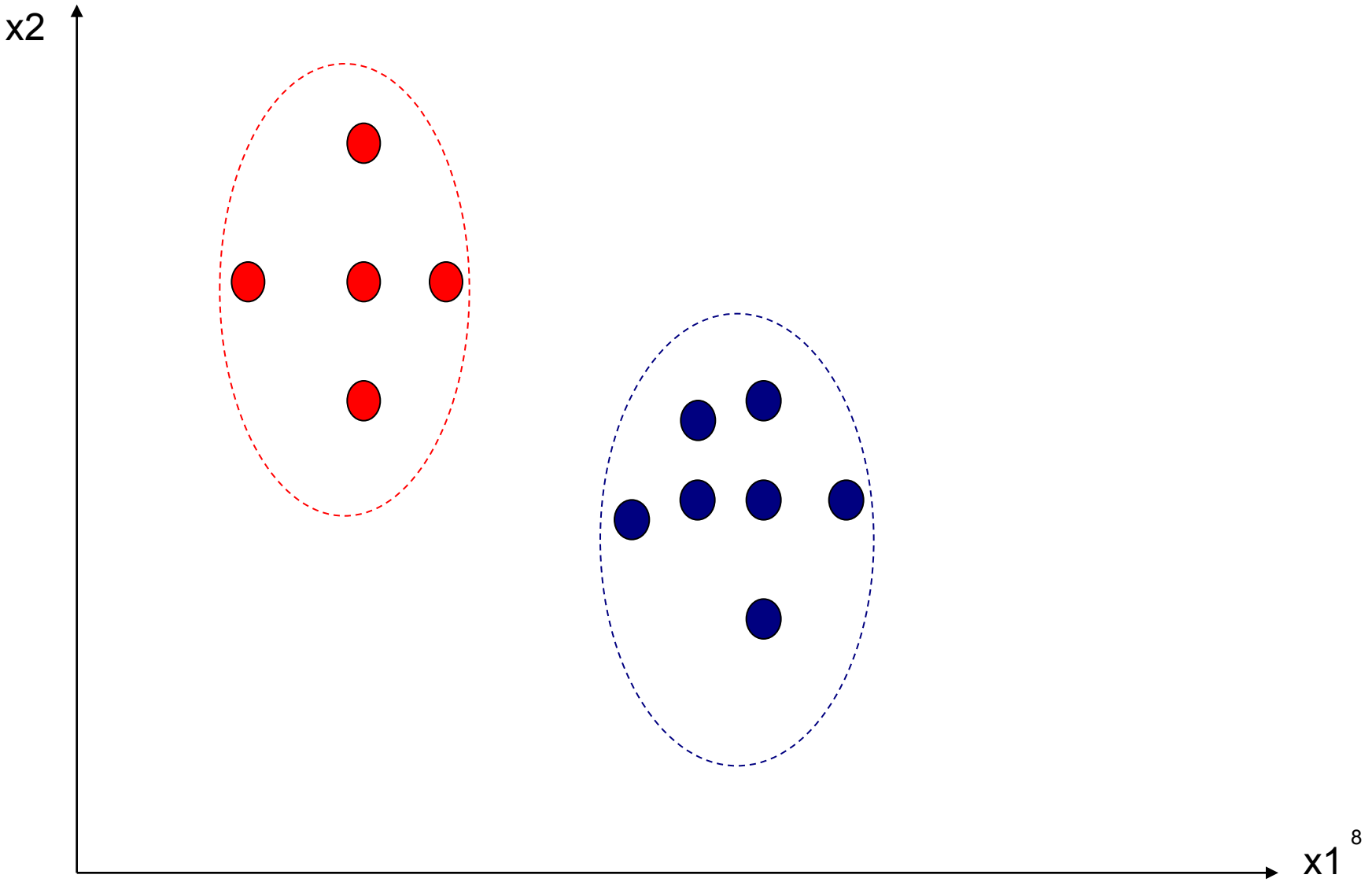
Semi-supervised



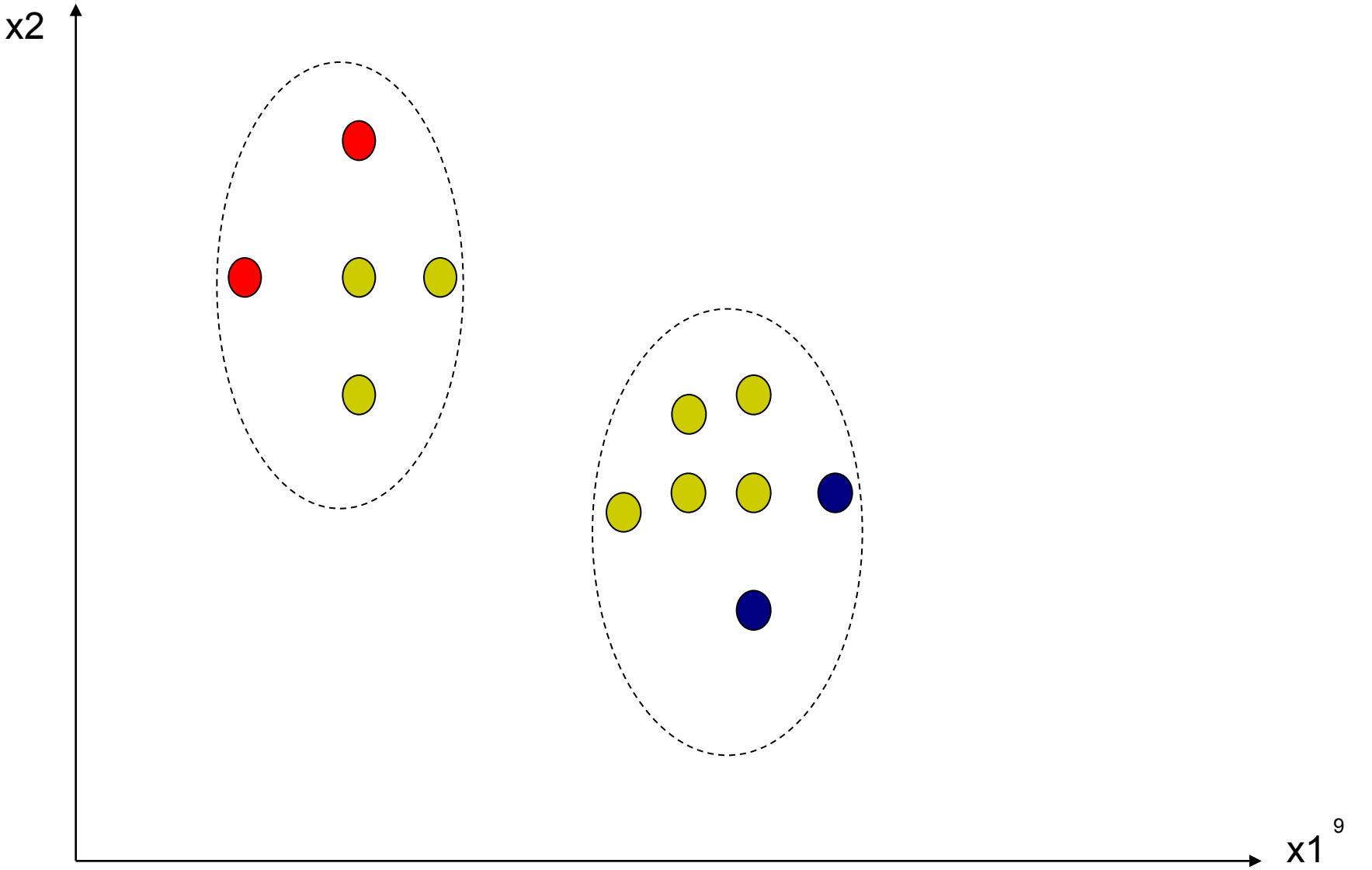
Unlabeled data → usupervised learning



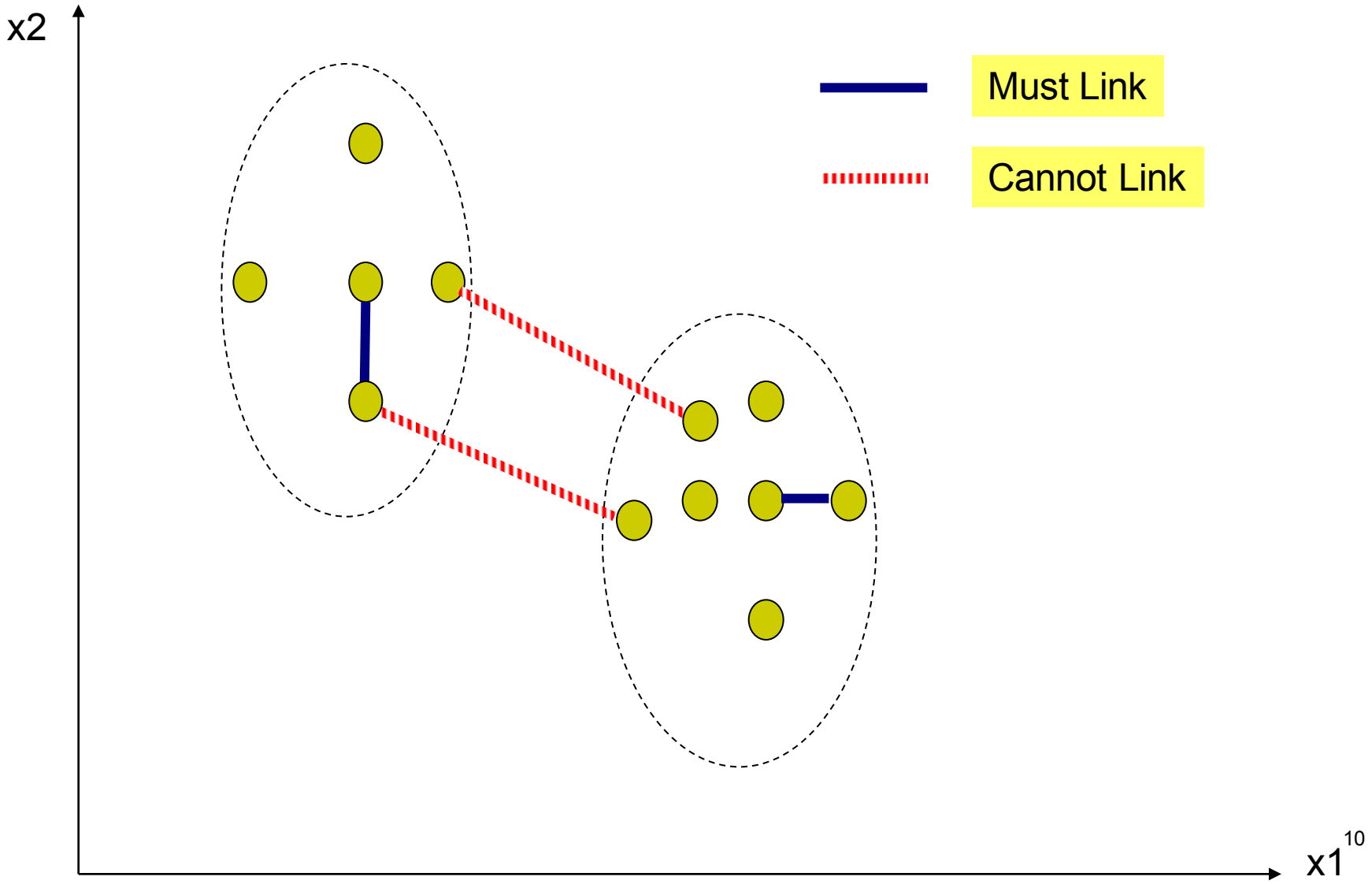
Labeled data → supervised learning



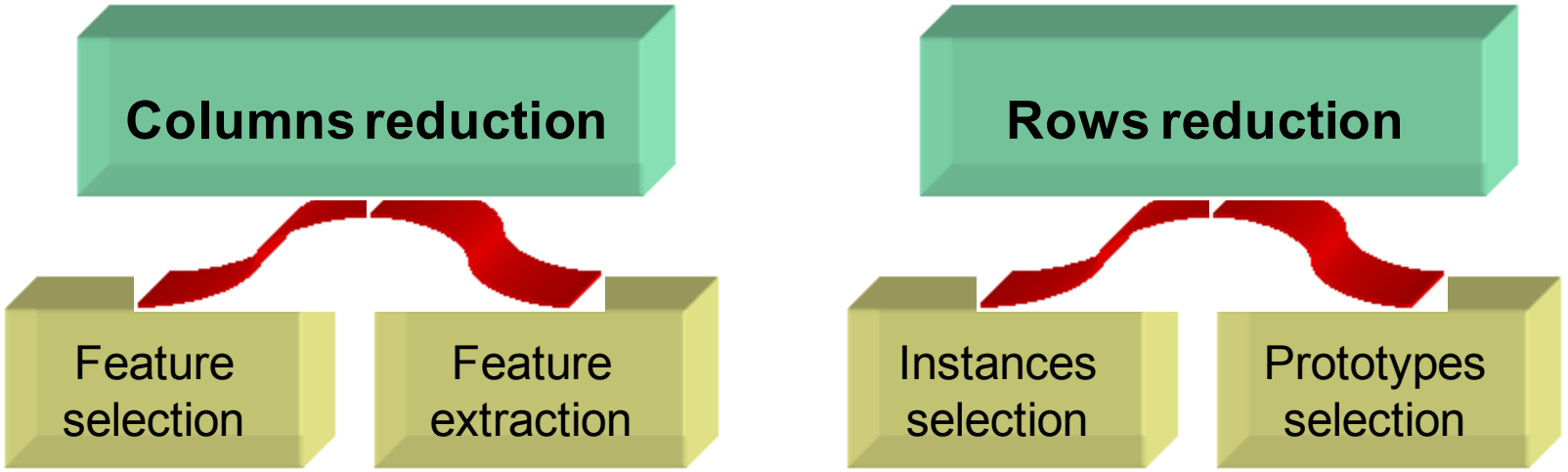
Partially labeled data → semi-supervised learning



Pairwise constraints data → semisupervised learning



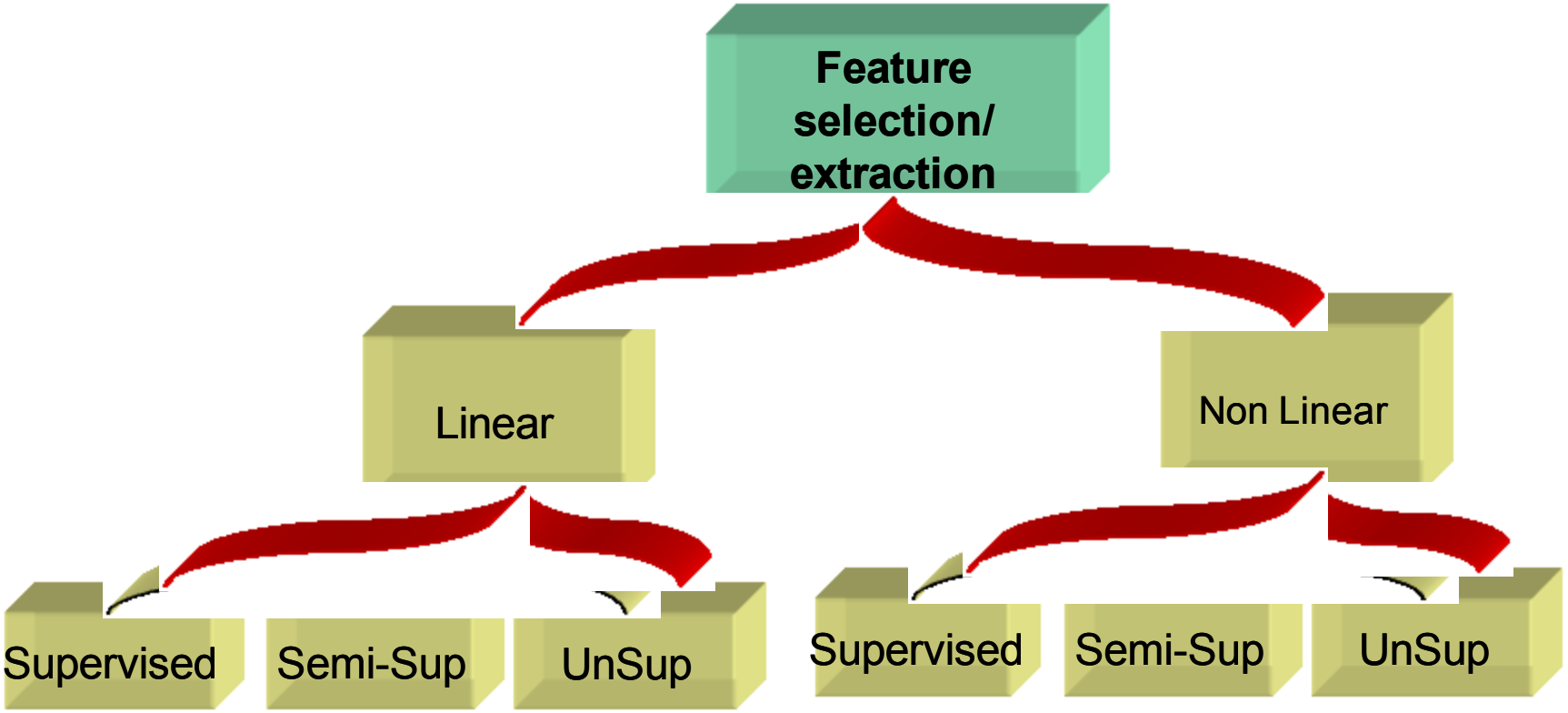
Feature selection / extraction approaches



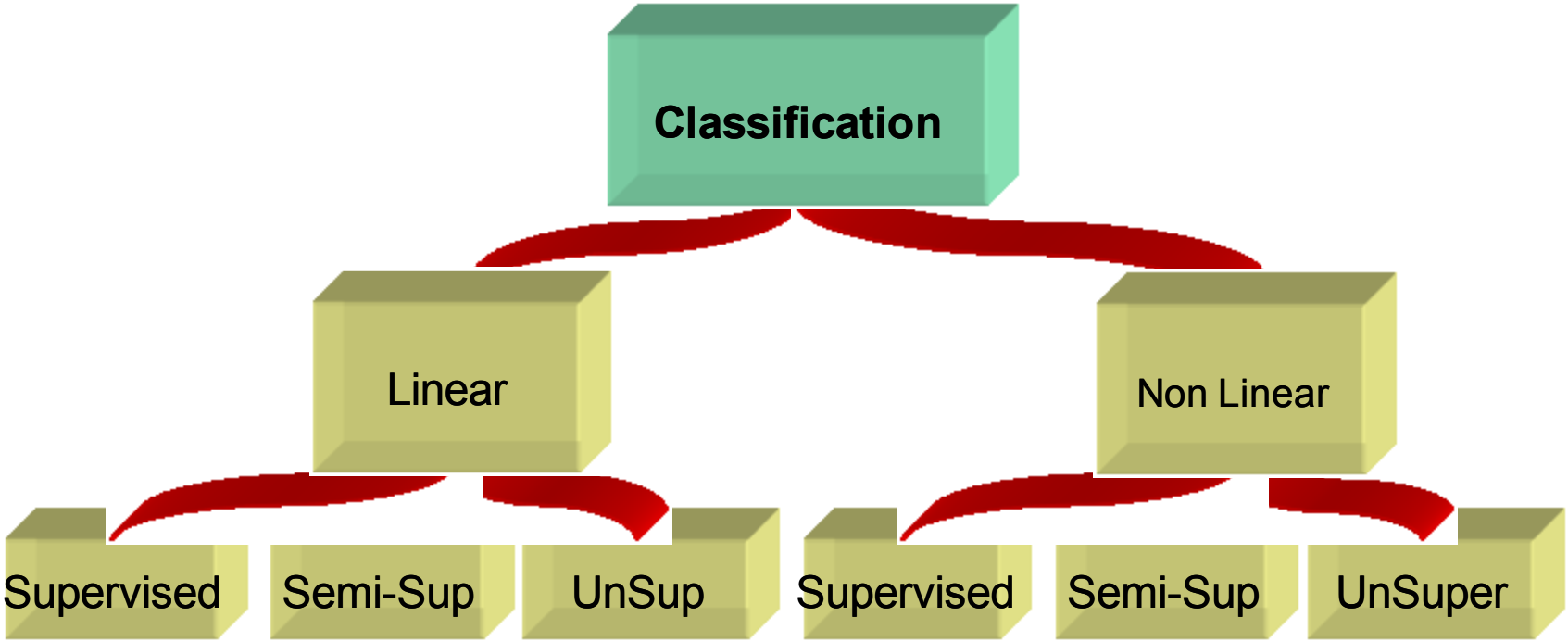
$\{a_1, \ldots, a_l\} \rightarrow \{b_1, \ldots, b_m\}$ $\quad l < m$

$\{X_1, \ldots, X_n\} \rightarrow \{W_1, \ldots, W_k\}$ $\quad k < n$

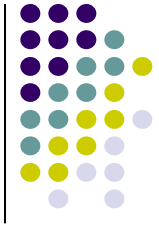
Feature selection / extraction approaches



Classification approaches



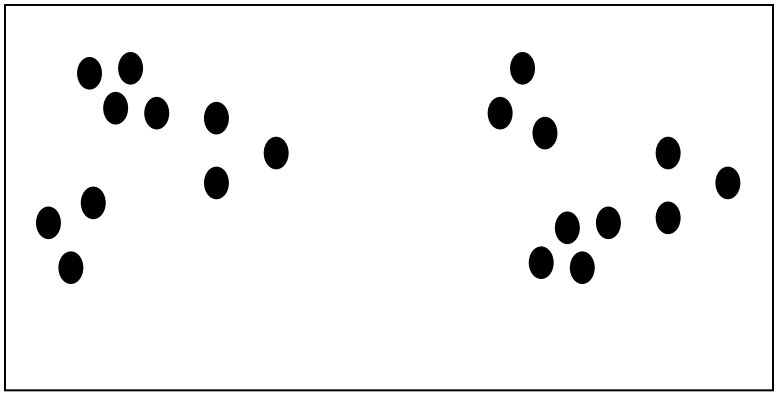
Data Clustering



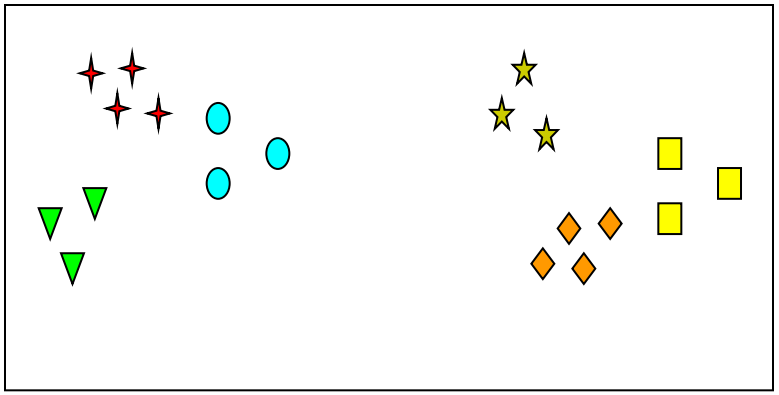
- Data clustering is an important problem with many applications in:
  - Machine learning,
  - Computer vision,
  - Signal processing…

- The object of clustering is to divide a dataset into natural groups such as:
  - Points in the same group are similar
  - Points in different groups are dissimilar to each other.
- Clustering methods can be:
  - Hierarchical: Single Link, Complete Link, etc.
  - Partitional or flat: k-means, Gaussian Mixture, Mode Seeking, Graph partitioning, etc.

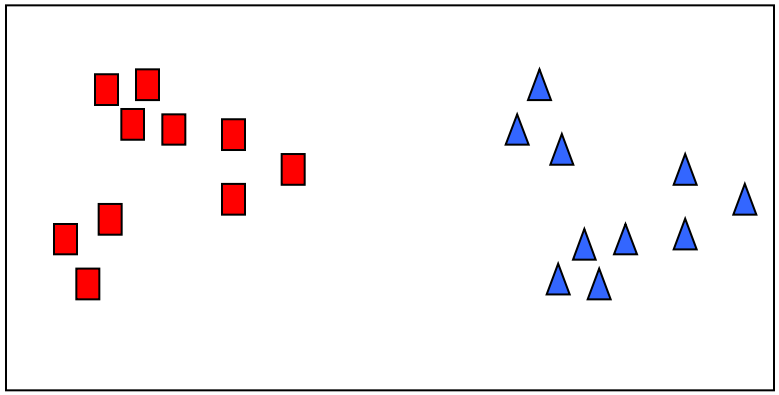
How many clusters?



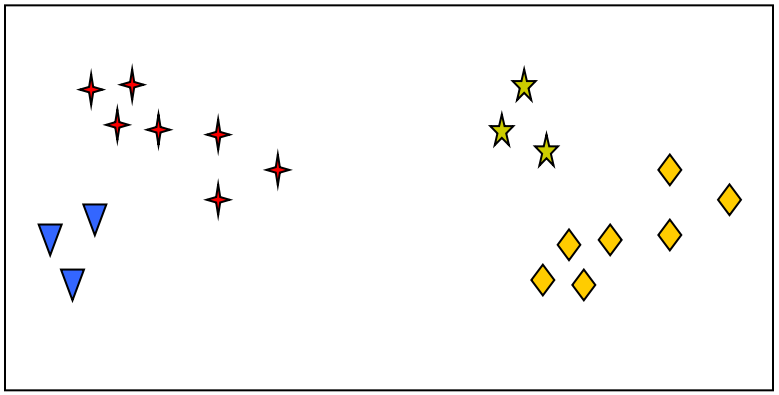
How many clusters?



Six?

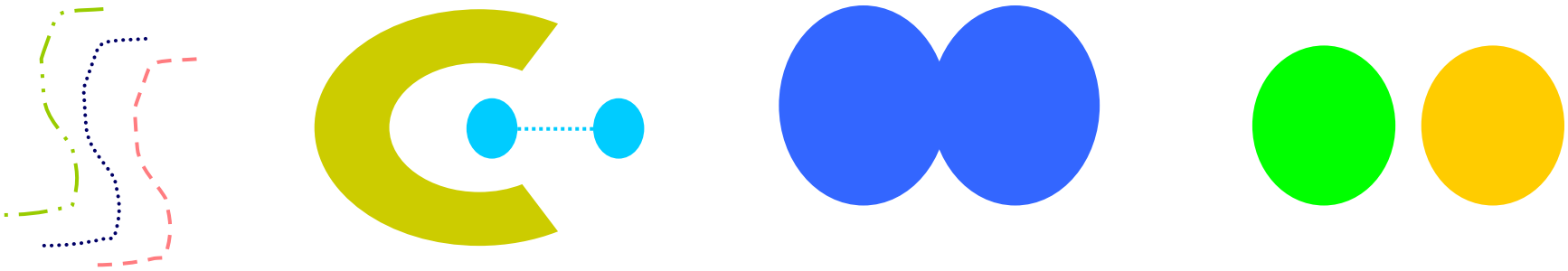


Two?

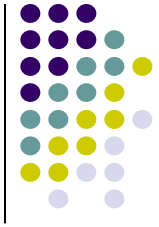


Four?

Clusters forms

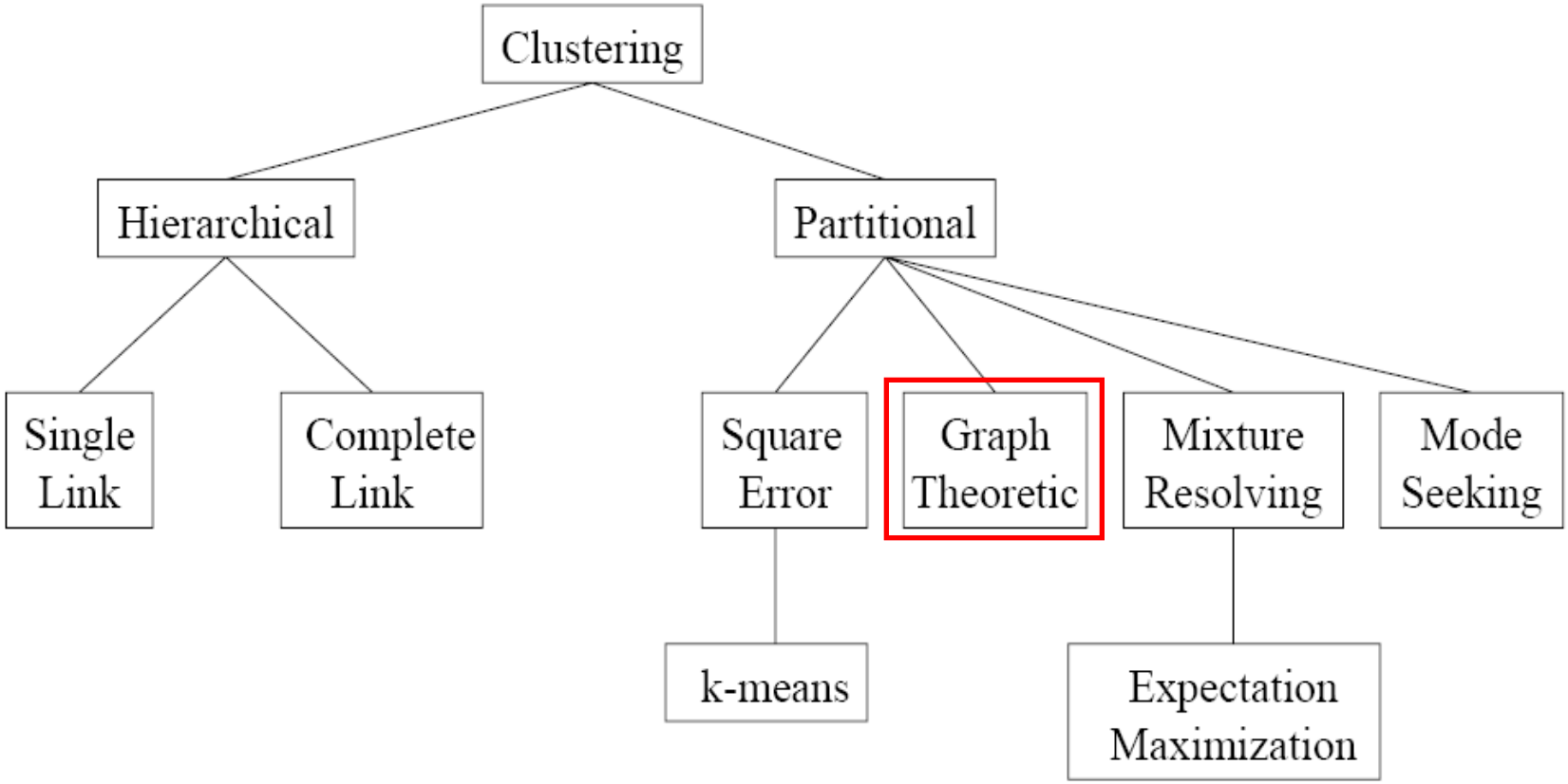


Clustering assumptions

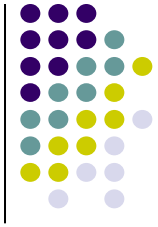


- Clustering algorithms are based on implicit assumptions about the definition of cluster's structure.
- Generally, a cluster can be defined as a set of points that share some property:
  - well-separated: A cluster is a set of points in which each point is closer to every other point in the cluster than to any point not in the cluster.
  - prototype-based: A cluster is a set of points in which is point is closer to the prototype that define the cluster than the prototype of any other cluster. (K-means, K-medoïds),
  - density based: A cluster is a dense region of point that is surrounded by a region of low density. (mixture models)
  - graph-based: A cluster is a group of points connected to one another. (spectral clustering).

Clustering methods



Graph Theory View of Clustering



- The database is composed of n points:

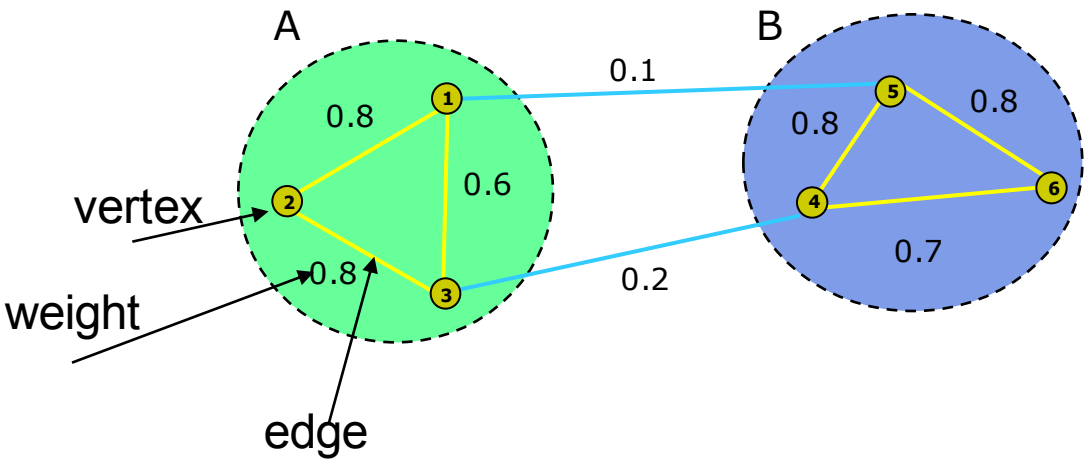
  $\chi = \{x_1, \ldots, x_i, \ldots, x_n\} \quad x_i \in R^{\ell}$

- Points are characterized by their pair-wise similarities i.e. to each pair of points (x_i, x_j) is associated a similarity value w_{ij}. Similarity matrix $W \in R^{nxn}$ is then composed of terms w_{ij} which can be of: $w_{ij} = f\left(d(x_i, x_j); \theta\right)$
  - Cosine,
  - Fuzzy,
  - Gaussian types.
- Gaussian type is much more used in clustering approaches and is defined by:

$$w_{ij} = \exp\left(-\frac{1}{2\sigma^2} d^2(x_i, x_j)\right)$$

Illustrative example

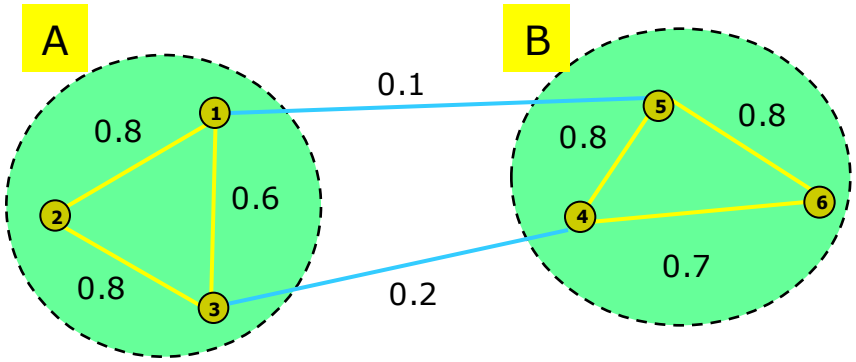
- Dataset is composed of six points: $\{x_1, x_2 ..., x_6\}$



Illustrative example

- We define the degree d_i of a vertex i as the sum of edges weights incident to it:

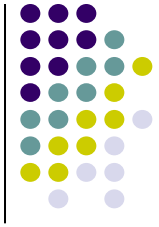
$$d_i = \sum_{j=1}^{n} w_{ij}$$



$d_1 = 0.8 + 0.6 + 0.1$

$d_1 = 1.5$

Graph cut



- The degree matrix of the graph G denoted by "D" will be a diagonal matrix having elements d_i on its diagonal and the off-diagonal elements having value 0.
- Given two disjoint clusters (subgraphs) A and B of the graph G, we define the following three terms:
  - The sum of weight connections between two clusters:

$$\mathrm{Cut}(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

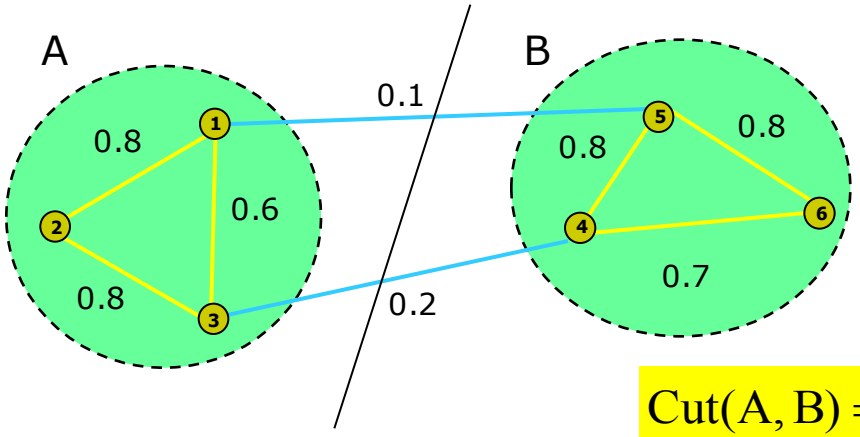
  - The sum of weight connections within cluster A:

$$\mathrm{Cut}(A,A) = \sum_{i \in A, j \in A} w_{ij}$$

  - The total weights of edges originating from cluster A.

$$\mathrm{Vol}(A) = \sum_{i \in A} d_i \qquad d_i = \sum_{j=1}^{n} w_{ij}$$

Graph cut



$$Cut(A,B) = \sum_{i \in A, j \in B} w_{ij}$$ ⇒ Cut(A, B) = 0.3

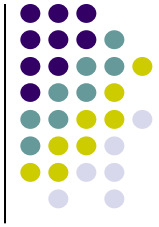
$$Cut(A,A) = \sum_{i \in A, j \in A} w_{ij}$$ ⇒ Cut(A, A) = 2.2

$$Cut(B,B) = \sum_{i \in B, j \in B} w_{ij}$$ ⇒ Cut(B, B) = 2.3

$$Vol(A) = \sum_{i \in A} \sum_{j=1}^{n} w_{ij} = \sum_{i \in A} d_i$$ ⇒ Vol(A) = 4.7

$$Vol(B) = \sum_{i \in B} \sum_{j=1}^{n} w_{ij} = \sum_{i \in B} d_i$$ ⇒ Vol(B) = 4.9

Minimum cut method



- The objective of MinCut method is to find two sets (clusters) A and B which have the minimum weight sum connections. So the objective function of this method is simple and defined by:

$$J_{MinCut} = Cut(A, B)$$

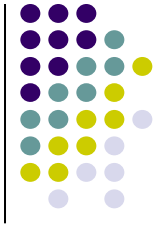
- It is easy to prove that such equation can be written as:

$$J_{MinCut} = \frac{1}{4} u^T (D - W) u$$

- $q_i \in R^n$ is the indicator vector of vertices belonging to clusters A and B such that:

$$q_i = \begin{cases} +1 & i \in A \\ -1 & i \in B \end{cases}$$

Minimum cut method



- When relaxing indicator vector from binary to continuous values in an interval, the solution minimizing the objective function will be equivalent to solve the following equation:

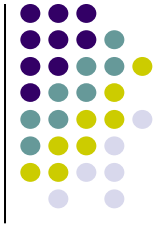
$$(D - W)u = \lambda u$$

- The Laplacian matrix L is defined by:

$$L = D - W$$

- Laplacian matrix L presents a trivial solution given by the eigenvalue "0" and eigenvector "e": e = (1, 1, …, 1)^T.
- The second smallest eigenvector , also called Fiedler vector, will be used to bi-partition of the graph by finding the optimal splitting point.
- In this method there is no mention of the cluster size and experiments showed that it works only when clusters are balanced and there are no isolated points.

Normalized and MinMax cut methods



- 1st constraint: inter-connections should be minimized:

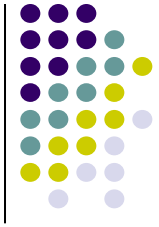
  cut(A, B) minimum
- 2nd constraint: intra-connections should be maximized:

  cut(A, A) and cut(B, B) maximum
- These requirements are simultaneously satisfied by minimizing these objective functions

$$J_{NCut}(A,B) = Cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{Vol(B)}\right)$$

$J_{NCut}(A,B) = 0.125$

Normalized and MinMax cut methods



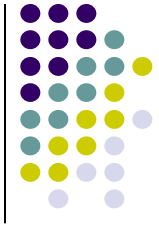
- By relaxing the indicator vector “u" to real values, it is proved that, minimizing NCut objective function is obtained by the second smallest eigenvector of the generalized eigenvalue system:

$$(D - W)y = \lambda Dy$$

$$L_{NCut} = D^{-1/2}(D - W)D^{-1/2}$$

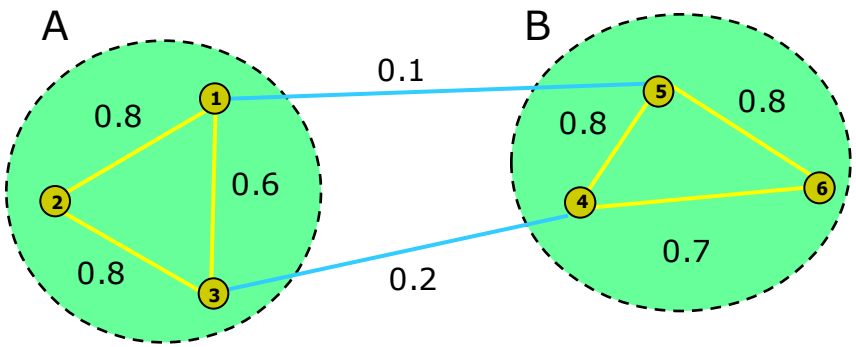
- Similar procedure can be also applied to MinMaxCut method

Spectral clustering steps

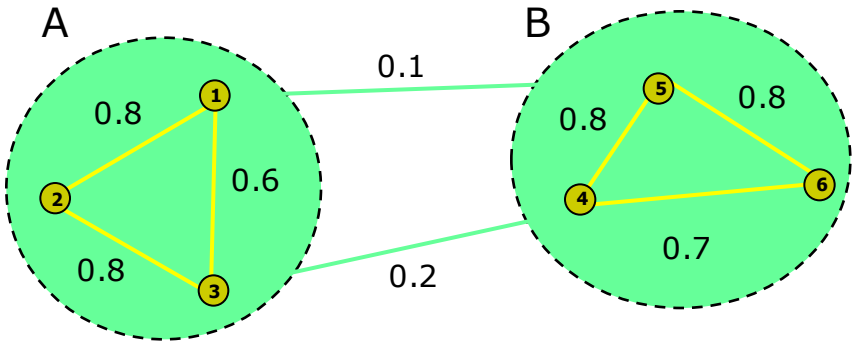


- **Pre-processing**
  - Construct the graph and the similarity matrix representing the dataset.
- **Spectral representation**
  - Form the associated Laplacian matrix
  - Compute eigenvalues and eigenvectors of the Laplacian matrix.
  - Map each point to a lower-dimensional representation based on one or more eigenvectors.
- **Clustering**
  - Assign points to two or more classes, based on the new representation.

Illustrative example



Graph and similarity matrix

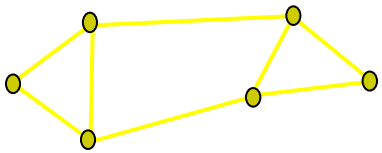


	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0.8	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

Illustrative example

Pre-processing

Build Laplacian matrix L of the graph



x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	-0.8	0	-0.2	2.5	-0.8	-0.7
x_5	-0.1	0	0	0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

Decomposition: Find

- eigenvalues Λ and
- eigenvectors X of matrix L

$\Lambda =$

0.0
0.3
2.2
2.3
2.5
3.0

$X =$

0.4	0.1	0.4	-0.2	-0.9
0.4	0.1	-0.	0.4	0.3
0.4	-0.2	0.0	-0.2	0.6
0.4	0.9	0.2	-0.4	-0.6
0.4	-0.4	-0.8	-0.6	-0.2
0.4	-0.2	0.5	0.8	0.9

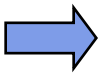
- Map vertices to the corresponding components of 2nd eigenvector

x_1	0.2
x_2	0.2
x_3	0.2
x_4	-0.4
x_5	-0.7
x_6	-0.7

How do we find the clusters?

Spectral Clustering Algorithms (continued)

x_1	0.2
x_2	0.2
x_3	0.2
x_4	-0.4
x_5	-0.7
x_6	-0.7



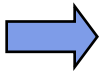
Split at value 0

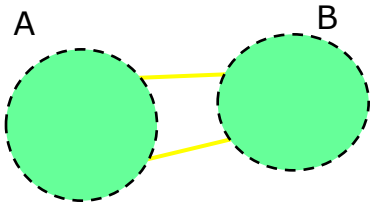
Cluster A: Positive points

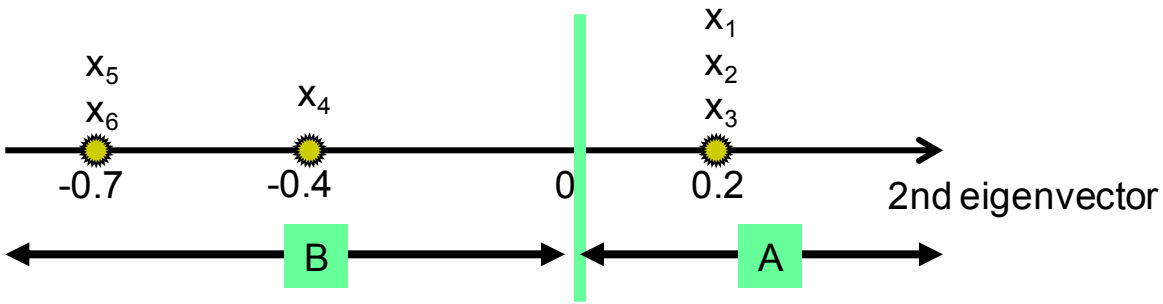
Cluster B: Negative points

x_1	0.2
x_2	0.2
x_3	0.2

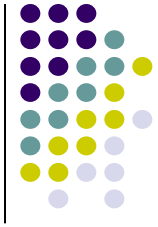
x_4	-0.4
x_5	-0.7
x_6	-0.7







k-way graph cuts



In order to partition a dataset or graph into k classes, two basic approaches can be used:

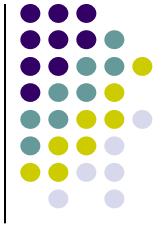
- Recursive bi-partitioning: The basic idea is to recursively apply bi-partitioning algorithm in a hierarchical way: after partitioning the graph into two, reapply the same procedure to the subgraphs. The number of groups is supposed to be given or directly controlled by the threshold allowed to the objective function.
- k-way partitioning: The 2-way objective functions can be generalized to take into consideration more than two clusters (Card(A_i) = $|A_i|$):

$$J_{RatioCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A}_i)}{|A_i|}$$

$$J_{NCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A}_i)}{Vol(A_i)}$$

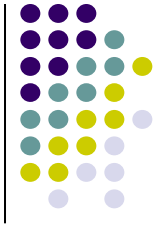
$$J_{MinMaxCut}(A_1,..,A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A}_i)}{Cut(A_i, A_i)}$$

Spectral clustering steps



- **Pre-processing**
  - Construct the graph and the similarity matrix representing the dataset.
- **Spectral representation**
  - Form the associated Laplacian matrix
  - Compute eigenvalues and eigenvectors of the Laplacian matrix.
  - Map each point to a lower-dimensional representation based on one or more eigenvectors.
- **Clustering**
  - Assign points to two or more classes, based on the new representation.

NJW algorithm



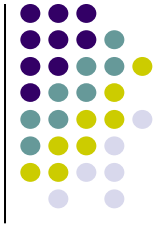
Given a set of points that we want to partition into k clusters: $\chi = \{x_1, \cdots, x_n\}$

1. Form the similarity matrix W defined by: $w_{ij} = \exp\left(-\frac{1}{2\sigma^2} d^2(x_i, x_j)\right)$

2. Construct the Laplacian matrix: $L_{NCut} = D^{-1/2}(D - W)D^{-1/2}$

3. Find the k first eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix U by stacking the eigenvectors in columns: $U = [u_1 \vdots \cdots \vdots u_k] \in R^{nxk}$
4. Form the matrix Y from U by normalizing each of U's rows to have unit length:
5. Treat each row of Y as a point in R^k and classify them into k classes via k-means or any other algorithm: $Y_{ij} = \frac{U_{ij}}{[\sum_{j=1}^{n} U_{ij}^2]^{1/2}}$
6. Assign the original points x_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Variants of spectral clustering algorithms

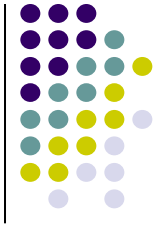


- In general, the spectral clustering methods can be divided into three main varieties since the basic spectral algorithm is itself divided to three steps:
- Preprocessing: Spectral clustering methods can be best interpreted as tools for analysis of the block structure of the similarity matrix. So, building such matrices may certainly ameliorate the results.
  - Calculation of the similarity matrix is not evident.
  - Choosing the similarity function can highly affect the results of the following steps. In most cases, the Gaussian kernel is chosen, while other similarities like cosine similarity are used for specific applications.
- Graph and similarity matrix construction: Laplacian matrices are generally chosen to be positive and semi-definite thus their eigenvalues will be non-negatives. The most used Laplacian matrices are summarized in the following.

Unnormalized	$L = D - W$
Symmetric	$L_{Sy} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$
Asymmetric	$L_{As} = D^{-1} L = I - D^{-1} W$

- Clustering: simple algorithms other than k-means can be used in the last stage such as simple linkage, k-lines, elongated k-means, mixture model, etc.

Parameters tuning (1/2)



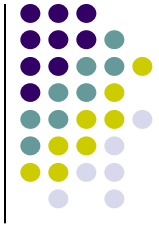
The free parameter of Gaussian kernel is often overlooked. Indeed, with different values of σ, the results of clustering can be vastly different:

- The parameter is done manually.
- The parameter is selected automatically by running the spectral algorithm repeatedly for a number of values and selecting the one which provides least distorted clusters in spectral representation space.

Weakness of this method:

- computation time,
- the range of values to be tested has to be set manually
- with input data including clusters with different local statistics there may not be a single value of that works well for all the data.

Parameters tuning (2/2)



- Local scaling parameter: A local scaling parameter for each data point is calculated:

$$d^2(x_i, x_j) \Rightarrow \frac{d(x_i, x_j)\, d(x_j, x_i)}{\sigma_i \sigma_j}$$

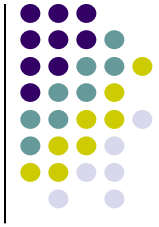
$$w_{ij} = \exp\left(\frac{-1}{\sigma_i \sigma_j} d^2(x_i, x_j)\right)$$

- The selection of the local scale can be done by studying the local statistics of the neighborhood of point . For example, it can be chosen as:

$$\sigma_i = d(x_i, x_m)$$

where x_m is the m-th neighbor of point x_i. The selection of "m" is independent of the scale.

Estimating of the number of clusters (1/2)



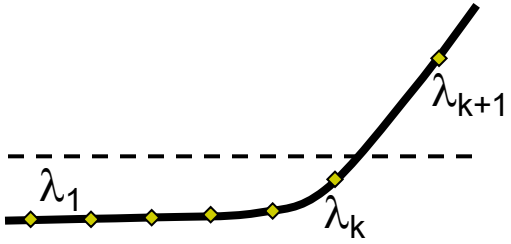
The main difficulty of clustering algorithms is the estimation of the number of clusters.

- Number of clusters is set manually.
- Number of clusters is automatically discovered:
  - eigengap detection
  - canonical coordinates

1- Eigengap detection: Find the number of clusters by analyzing the eigenvalues of the Laplacian matrix,

- The number of eigenvalues of magnitude 0 is equal to the number of clusters k. This implies one could estimate k simply by counting the number of eigenvalues equaling 0. This criterion works when the clusters are well separated,
- Search for a drop in the magnitude of the eigenvalues arranged in increasing order,
- Here, the goal is to choose the number k of clusters such that all eigenvalues are very small $\lambda_1, \lambda_2, \ldots, \lambda_k$ are very small while λ_{k+1} is relatively large.

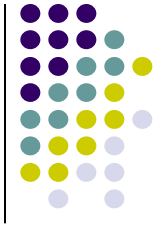
Estimating the number of clusters (2/2)



2- Find canonical coordinate system:

- Minimize the cost of aligning the top eigenvectors with a canonical coordinate system,
- The search can be performed incrementally,
- At each step of the search, a single eigenvector is added to the already rotated ones,
- This can be viewed as taking the alignment results of the previews number as an initialization to the current one.

Computation and memory problems



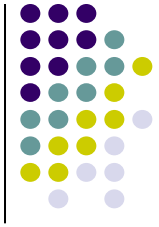
In signal and image processing domains, similarity matrix grows as the square of the number of elements in the dataset, it quickly becomes infeasible to fit in memory. Then, the need to solve eigensystem presents a serious computational problem.

Solutions:

- Sparse similarity matrix: One approach is to use a sparse, approximate version of similarity in which each element is connected to only a few of its nearby neighbors and all other connections are assumed to be zero.
  - Employ efficient eigensolvers: Lanczos iterative approach.

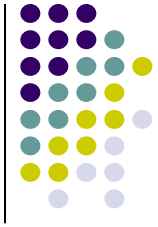
- Nyström method: Numerical solution of eigenfunction problem. This method allows one to extrapolate the complete grouping solution using only a small random number of samples. In doing so, we leverage the fact that there are far fewer coherent groups in a scene than pixels.

Conclusion



- Some spectral clustering methods are presented.
- They can be divided into three categories according to the basic stages of standard spectral clustering:
  - pre-processing,
  - spectral representation,
  - clustering.
- We pointed out various solutions to their main problems:
  - parameters tuning,
  - number of clusters estimation,
  - complexity computation.
- The success of spectral methods is due to their capacity to discover the data clusters without any assumption about their statistics.
- For these reasons, they are recently applied in different domains particularly in signal and image processing.

Publications

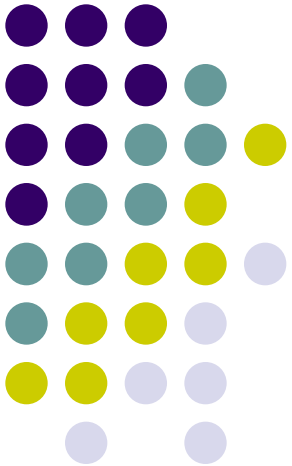


- Mariam Kalakech, Philippe Biela, Ludovic Macaire, Denis Hamad: *Constraint scores for semi-supervised feature selection*: A comparative study. **Pattern Recognition Letters** 32(5): 656-665 (2011).
- Mariam Kalakech, Philippe Biela, Denis Hamad, Ludovic Macaire: *Constraint Score Evaluation for Spectral Feature Selection*. **Neural Processing Letters**. DOI 10.1007/s11063-013-9280-2. Online January 2013.
- G. Wacquet, É. Poisson Caillault, D. Hamad, P.-A. Hébert: Constrained spectral embedding for K-way data clustering. **Pattern Recognition Letters** 34 (2013) 1009–1017.

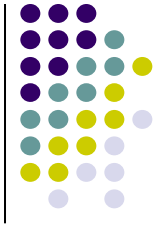
Bibliography

- Bach F. R. and M. I. Jordan, “Learning spectral clustering, with application to speech separation”, Journal of Machine Learning Research, vol. 7, pp. 1963-2001, 2006.
- Bach F. R. and M. I. Jordan, “Learning spectral clustering”, In Thrun S. and Soul L. editors. NIPS’16, Cambridge, MA, MIT Press, 2004.
- Chang H., D.-Y. Yeung, “Robust path-based spectral clustering”, Pattern Recognition 41 (1), pp. 191-203, 2008.
- Ding C., X. H. He, Zha, M. Gu, and H. Simon, “A min-max cut algorithm for graph partitioning and data clustering”. IEEE first Conference on Data Mining, pp. 107-114, 2001.
- Fowlkes C., S. Belongie, F. Chung, and J. Malik, “Spectral Grouping Using the Nyström Method”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pp. 214-225, 2004.
- Hagen L. and A. Kahng, “Fast spectral methods for ratio cut partitioning and clustering”. In Proceedings of IEEE International Conference on Computer Aided Design, pp. 10-13, 1991.
- Jain A., M. N. Murty and P. J. Flynn “Data clustering: A review”. ACM Computing Surveys, vol. 31(3), pp. 264-323, 1999.
- Meila M. and L. Xu, “Multiway cuts and spectral clustering”, Advances in Neural Information Processing Systems, 2003.
- Ng, A. Y., M. I., Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm”. Advances in Neural Information Processing Systems 14, volume 14, pp. 849-856, 2002.
- Sanguinetti G., J. Laidler, and N. D. Lawrence. "Automatic determination of the number of clusters using spectral algorithms", IEEE Machine Learning for Signal Processing conference, Mystic, Connecticut, USA, 28-30 september, 2005.
- Shi. J. and J. Malik. "Normalized cuts and image segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no 8, pp. 888-905, 2000.
- von Luxburg U., “A tutorial on spectral clustering”, Technical report, No TR-149, Max-Planck-Institut für biologische Kybernetik, 2007.
- Xiang T. S. Gong, “Spectral clustering with eigenvector selection”, Pattern Recognition, vol. 41, no 3, pp. 1012-1029, 2008.
- Yu S.X. and J. Shi “Multiclass spectral clustering” 9th IEEE International Conference on Computer Vision CCV, Washinton, DC, USA, 2003.
- Yu S.X., J. Shi, Segmentation given partial grouping constraints, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no 2, pp. 173–183, 2004.
- Zelnik-Manor L., P. Perona, “Self-Tuning spectral clustering”, Advances in Neural Information Processing Systems, vol. 17, 2005.
- Derek greene, "Graph partitioning and spectral clustering", https://www.cs.tcd.ie/research_groups/mlg/kdp/presentations/Greene_MLG04.ppt

Semi-supervised dimensionality reduction

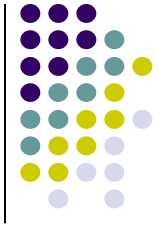


Introduction



- In pattern recognition and machine learning domains there are rapid accumulation of high dimensional data:
  - Digital images
  - Financial time series
  - Genes expression micro-arrays
- Dimensionality reduction is a fundamental tool in these domains.

Dimensionality reduction (1)

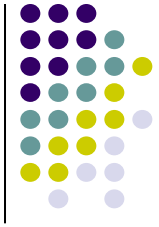


There are two types of dimensionality reduction:

- Feature selection consists to retain relevant features which constitute a low dimensional feature space.

- Feature extraction consists to transform the original representation space into a new low dimensional space by combining the initial features.

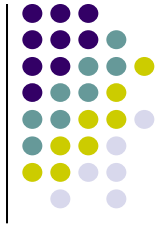
Dimensionality reduction (2)



- Unsupervised feature selection measures the feature capacity of keeping the intrinsic data structure in order to evaluate its relevance.

- Supervised feature selection consists in evaluating feature relevance by measuring the correlation between the feature and class labels.
- Supervised feature selection requires sufficient labeled data samples.
  - However, the sample labeling process by the user is fastidious and expensive.
  - That is the reason why in many real applications we are facing huge unlabeled data and small labeled samples: ”lack labeled-sample problem”,
- We propose spectral graph theory in order to elaborate semi-supervised criteria for feature selection.

Feature selection scores (filters)



- Supervised scores: Fisher score

$$F_r = \frac{\sum_{\omega=1}^{c} n_\omega (\mu_{\omega r} - \mu_r)^2}{\sum_{\omega=1}^{c} n_\omega \sigma_{\omega r}^2}.$$

- Unsupervised scores: Variance

$$V_r = \frac{1}{n} \sum_{i=1}^{n} (x_{ir} - \mu_r)^2 .$$

- Semi-supervised scores: Laplacian score

$$L_r = \frac{\sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}}{\sum_i (x_{ir} - \overline{f}_r) D_{ii}} = \frac{\widetilde{f}_r^T L \widetilde{f}_r}{\widetilde{f}r^T D \widetilde{f}_r}$$

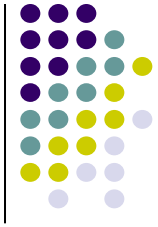
where $\widetilde{f}_r = f_r - \overline{f}_r$ and $\overline{f}_r = \frac{\sum_{i=1}^{n} x_{ir} d_i}{\sum_{i=1}^{n} d_i}$.

$$s_{ij} = \exp - \left(\frac{\|x_i - x_j\|^2}{2t^2} \right)$$

- Pairwise constraints scores

$$C_r^1 = \frac{\sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{M}}}{\sum_{ij} (x_{ir} - x_{jr})^2 s_{ij}^{\mathcal{C}}} = \frac{f_r^T L^{\mathcal{M}} f_r}{f_r^T L^{\mathcal{C}} fr}.$$

Semi-supervised dimensionality reduction

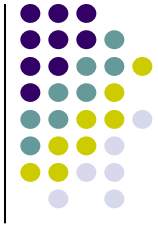


- Given:
  - Set of data samples: $X = \{x_1, \ldots, x_n\}$
  - Two types of pairwise constraints M and C:
    - Must-link pairwise constraints: $M = \{(x_i, x_j)\}$; $Card(M) = |M| = n_M$
    - Cannot-link pairwise constraints: $C = \{(x_i, x_j)\}$; $Card(C) = |C| = n_C$
- Problem:
  - Find $W = (w_1, \ldots, w_d)$ such that the transformed low-dimensional space defined by:

    $y_i = W^T X_i$

    Preserve the structure of the original data set as well as the pairwises constraints.

Objective function: Must and Cannot link constraints



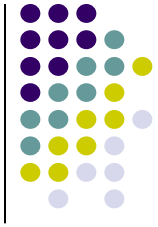
Objective function with Must-links and cannot links constraints:

$$J(w) = \frac{1}{2n_C} \sum_{(x_i, x_j) \in C} (y_i - y_j)^2 - \frac{\beta}{2n_M} \sum_{(x_i, x_j) \in M} (y_i - y_j)^2$$

$$J(w) = \frac{1}{2n_C} \sum_{(x_i, x_j) \in C} (w^T x_i - w^T x_j)^2 - \frac{\beta}{2n_M} \sum_{(x_i, x_j) \in M} (w^T x_i - w^T x_j)^2$$

The objective is to maximize the average squared distance in the transformed low-dimensional space between instances involved by the must and cannot links.

Objective function: Unlabelled data, Must and Cannot link constraints



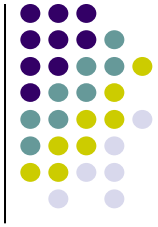
Objective function with Unlabelled data, Must and Cannot link constraints:

$$J(w) = \frac{1}{2n^2}\sum_{i,j}(w^T x_i - w^T x_j)^2 + \frac{\alpha}{2n_C}\sum_{(x_i,x_j)\in C}(w^T x_i - w^T x_j)^2 - \frac{\beta}{2n_M}\sum_{(x_i,x_j)\in M}(w^T x_i - w^T x_j)^2$$

Under the constraint $w^T w = 1$

The first term is the average squared distance between all data samples in the transformed space (PCA)

Objective function in spectral concept

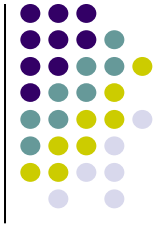


Objective function in spectral form with Unlabelled data, Must and Cannot link constraints:

$$S_{ij} = \begin{cases} \frac{1}{n^2} + \frac{\alpha}{n_C} & \text{if } (x_i, x_j) \in C \\ \frac{1}{n^2} - \frac{\beta}{n_C} & \text{if } (x_i, x_j) \in M \\ \frac{1}{n^2} & \text{otherwise} \end{cases}$$

$$J(w) = \frac{1}{2} \sum_{i,j} (w^T x_i - w^T x_j)^2 S_{ij}$$

Objective function in spectral concept



Objective function in spectral form with Unlabelled data, Must and Cannot link constraints:

$$S_{ij} = \begin{cases} \frac{1}{n^2} + \frac{\alpha}{n_C} & \text{if } (x_i, x_j) \in C \\ \frac{1}{n^2} - \frac{\beta}{n_C} & \text{if } (x_i, x_j) \in M \\ \frac{1}{n^2} & \text{otherwise} \end{cases}$$

$$\frac{1}{2}\sum_{i,j}(w^T x_i - w^T x_j)^2 S_{ij} = \frac{1}{2}\sum_{i,j}(w^T x_i x_i^T x + w^T x_j x_j^T w - 2w^T x_i x_j^T w)S_{ij}$$

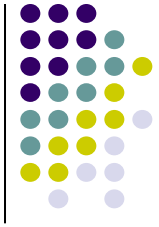
$$= \sum_{i,j} w^T x_i S_{ij} x_i^T w - \sum_{i,j} w^T x_i S_{ij} x_i^T w$$

$$= \sum_{i} w^T x_i D_{ii} x_i^T w - w^T X S X^T w$$

$$= w^T X (D - S) X^T w$$

$$= w^T X L X^T w$$

Problem formulation



- Given:
  - Set of data unlabelled samples: $X = \{x_1, \ldots, x_n\}$
  - Two types of pairwise constraints M and C:
    - Must-link pairwise constraints: $M = \{(x_i, x_j)\}$; $|M| = n_M$
    - Cannot-link pairwise constraints: $C = \{(x_i, x_j)\}$; $|C| = n_C$
- Problem:
  - Find $W = (w_1, \ldots, w_d)$ such that the transformed low-dimensional space defined by: $y_i = W^T X_i$

Under the constraint $w^T w = 1$.

$$J(w) = w^T X L X^T w$$