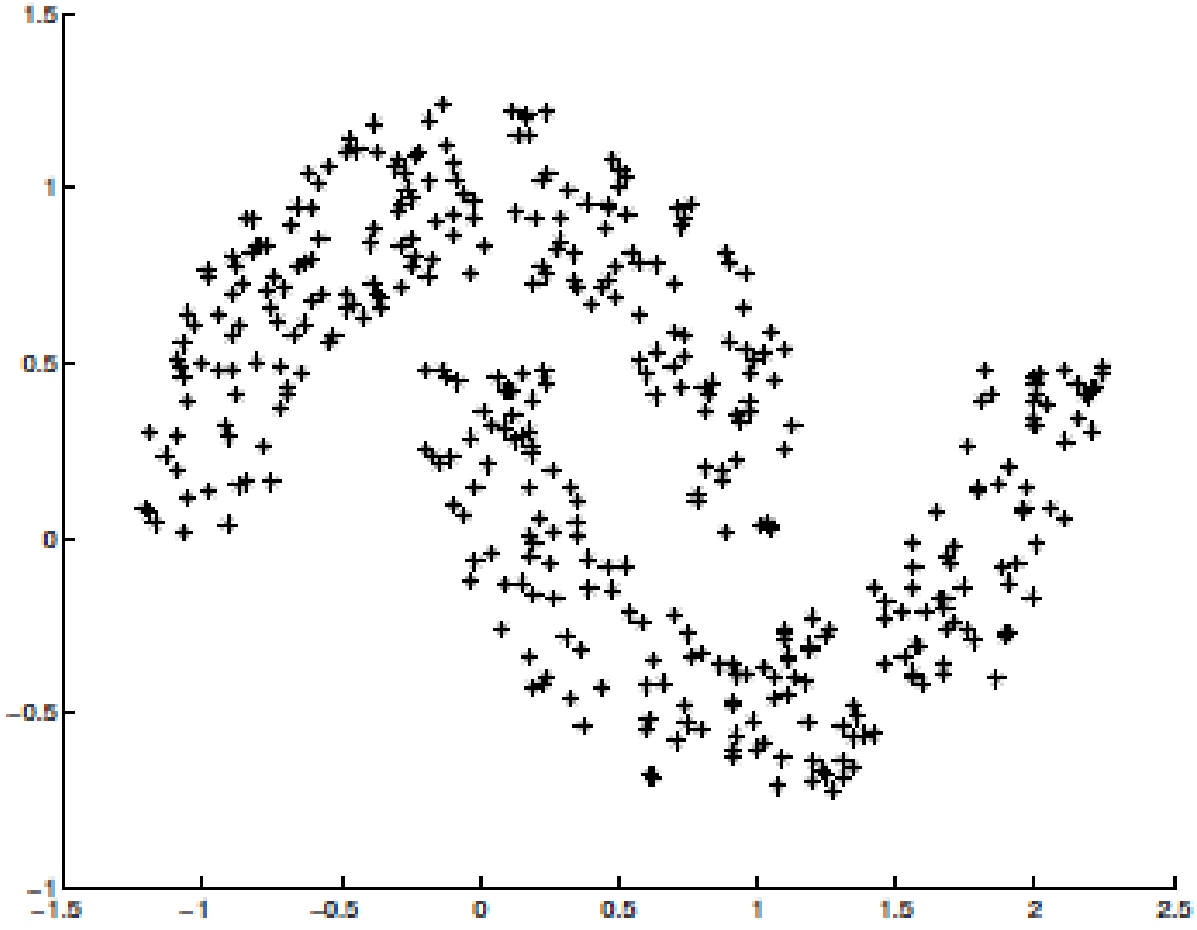
MANIFOLD LEARNING

Kavé Salamatian- LISTIC

Learning

- Machine Learning: develop algorithms to automatically extract ”patterns” or ”regularities” from data (generalization)
- Typical tasks
  - Clustering: Find groups of similar points
  - Dimensionality reduction: Project points in a lower dimensional space while preserving structure
  - Semi-supervised: Given labelled and unlabelled points, build a labelling function
  - Supervised: Given labelled points, build a labelling function
- All these tasks are not well-defined

Clustering

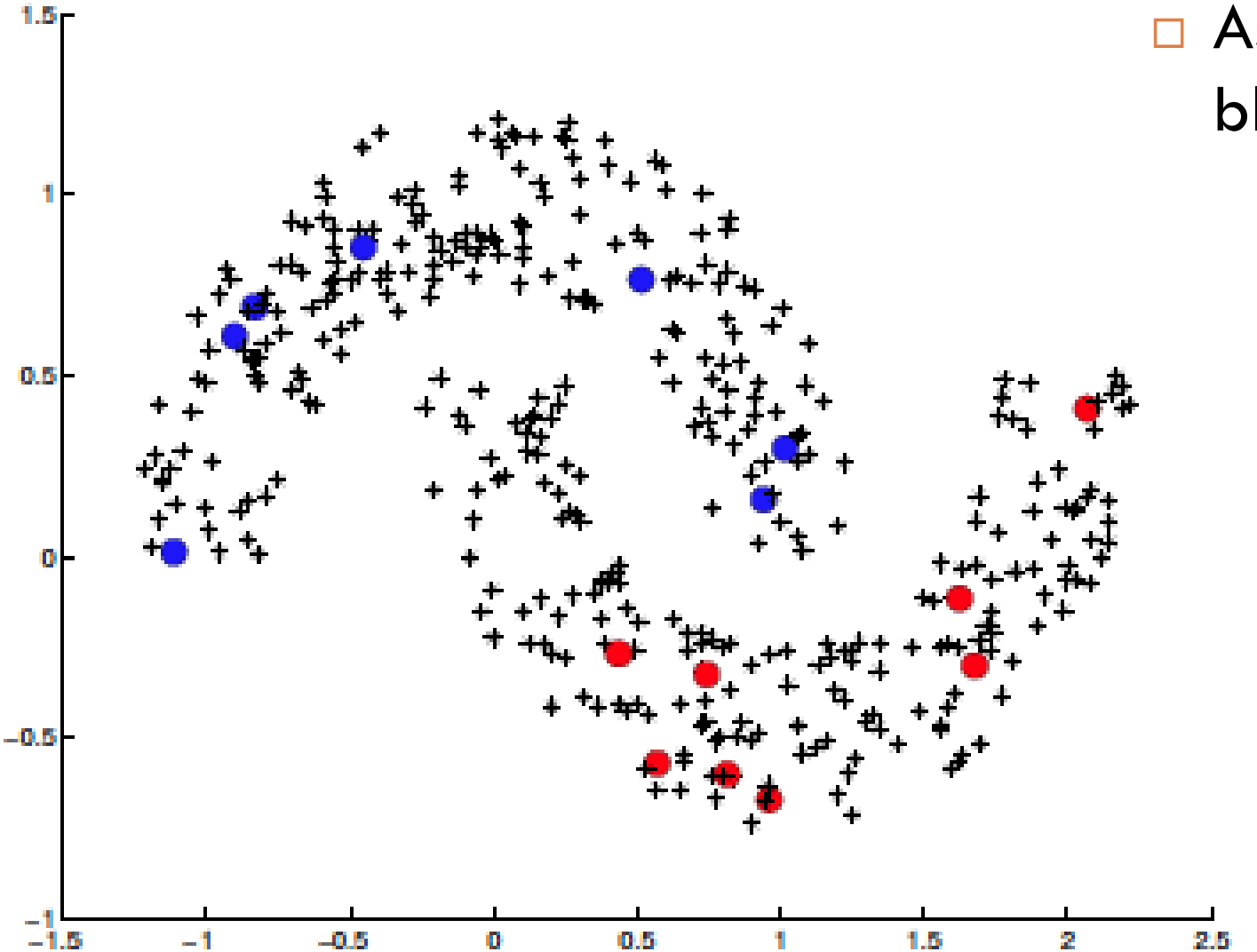


Identify the two groups

Dimensionality Reduction

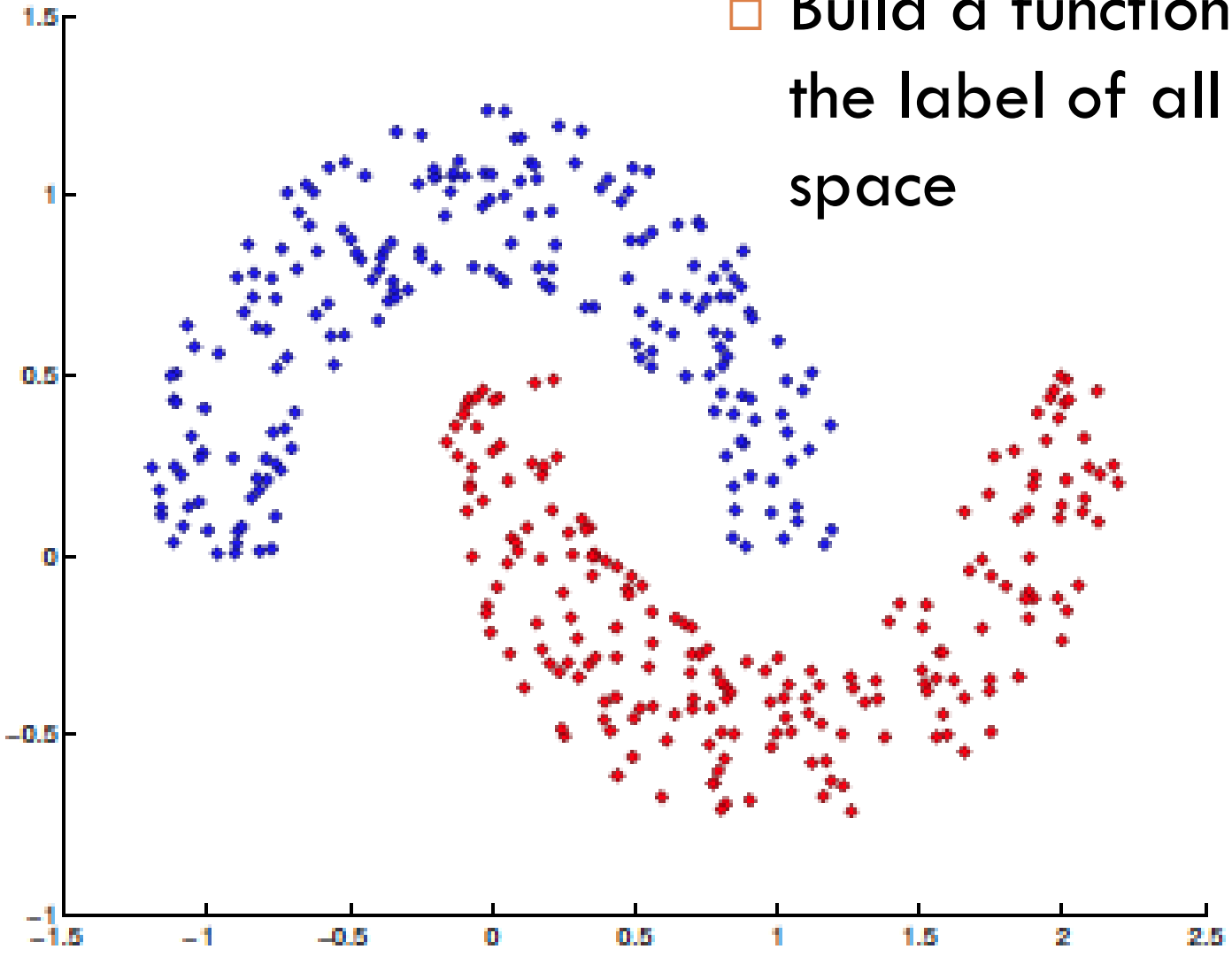
- Objects in R^{4096} ?
  - But only 3 parameters: 2 angles and 1 for illumination.
- They span a 3-dimensional sub-manifold of R^{4096}!

Semi-Supervised Learning



- Assign labels to black points

Supervised learning



- Build a function that predicts the label of all points in the space

Formal definitions

- Clustering: Given $\{X_1, \ldots, X_n\}$ build a function

$$f : \mathcal{X} \to \{X_1, \ldots, X_n\}$$

- Dimensionality reduction: Given $\{X_1, \ldots, X_n\} \in \mathbb{R}^D$ build a function $f : \mathbb{R}^D \to \mathbb{R}^d$
- Semi-supervised: Given $\{X_1, \ldots, X_n\}$ and $\{Y_1, \ldots, Y_m\}$ with $m << n$, build a function
- Supervised: Given $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, build $f : \mathcal{X} \to \mathcal{Y}$

Geometric learning

- Consider and exploit the geometric structure of the data
  - Clustering: two ”close” points should be in the same cluster
  - Dimensionality reduction: ”closeness” should be preserved
  - Semi-supervised/supervised: two ”close” points should have the same label
  - Examples: k-means clustering, k-nearest neighbors.

Manifold ?

- A manifold is a topological space that is locally Euclidian
  - Around every point, there is a neighborhood that is topologically isomorph to unit ball $\mathbb{R}^d$

Regularization

- Adopt a functional viewpoint for convenience
  - We want to define functions that conform with the regularities mentioned above:
    - values on close points should be close:

$$x_i \approx x_j \Rightarrow f(x_i) \approx f(x_j)$$

      - Gradient norm $\int ||\nabla f||^2 d\mu$
      - Weighted Gradient norm $\int ||\nabla f||^2 \alpha(\mu) d\mu$
      - Manifold Gradient $\int ||\nabla_{\mathcal{M}} f||^2 \alpha(\mu) d\mu$

Issues

- The manifold structure $\mathcal{M}$ is unknown, and the density are unknown!
  - Approaches:
    - density estimation
    - manifold estimation
    - density estimation on a manifold ???
  - None of these
    - Manifolds or densities may not exist as such
    - Just focus on the smoothness to be enforced on the data points

Intuition

- When the data has support on a low-dimensional submanifold the neighborhood graph is a discrete approximation of the submanifold
  - The neighborhood graph can be seen as a randomly sampled approximation of the continuous structure.
- In machine learning we are interested in the intrinsic properties and objects of this submanifold.
  - The graph Laplacian gives an approximation of the Laplace-Beltrami operator on the manifold

Spectral clustering

- Spectral clustering lies in spectral graph theory.
  - We consider the "similarity graph" induced by the data
    - Clustering reduces to the problem of graph partitioning:
      - we want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights (which means that points within the same cluster are similar to each
  - Different ways of formulating and solving the objective functions of such graph partitioning problems lead to normalized and non normalized spectral clustering

Interest of graph Laplacian

- The Laplacian is the generator of the diffusion process (label propagation in semi-supervised learning)
- The eigenvectors of the Laplacian have special geometric properties (motivation for spectral clustering),
- The Laplacian induces an adaptive regularization functional, which adapts to the density and the geometric structure of the data (semi-supervised learning, classification).

Spectral graph theory at a glance

- Study the properties of graphs via the eigenvalues and eigenvectors of their associated graph matrices
  - the adjacency matrix, the graph Laplacian and their variants.
    - These matrices have been extremely well studied from an algebraic point of view.
- The Laplacian allows a natural link between discrete representations (graphs), and continuous representations, such as metric spaces and manifolds.
- Laplacian embedding consists in representing the vertices of a graph in the space spanned by the smallest eigenvectors of the Laplacian
  - A geodesic distance on the graph becomes a spectral distance in the embedded (metric) space.

Spectral graph theory and manifold learning

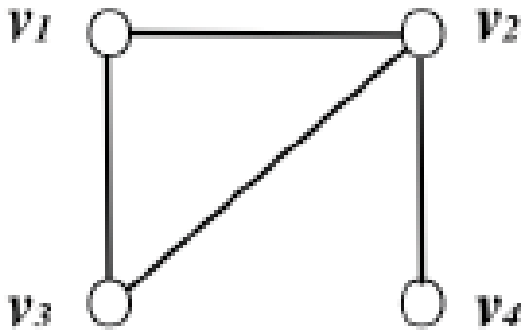
- First we construct a graph from $x_1, \ldots, x_n \in \mathbb{R}^D$
- we compute the d smallest eigenvalue-eigenvector pairs of the graph Laplacian
- We represent the data in the $\mathbb{R}^d$ space spanned by the corresponding orthonormal eigenvector basis.
  - Paradoxically, d may be larger than D

Adjacency matrix

- The adjacency matrix of a graph
  - For a graph with n vertices, the entries of the adjacency matrix are defined by:

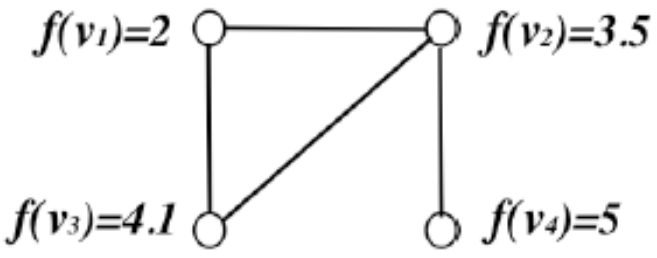
$$A = \begin{cases} a_{ij} = 1 & \text{if there is an edge } e_{ij} \\ a_{ij} = 0 & \text{if there is not an edge } e_{ij} \\ a_{ii} = 0 \end{cases}$$

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$



Real-valued functions on graphs

- We consider real-valued functions $f : \mathcal{V} \to \mathbb{R}$ on the set of the graph's vertices
  - Assigns a real number to each graph node.
    - Notation: $f = (f(v_1), \ldots, f(v_n)) = (f_1, \ldots, f_n) \in \mathbb{R}^n$
- The eigenvectors of the adjacency matrix, can be $\mathbf{A}X = \lambda X$ viewed as eigenfunctions.
  - Operator view $g = \mathbf{A}f, \; g(i) = \sum_{i \to j} f(j)$
  - Quadratic form $f^T \mathbf{A} f = \sum_{i \to j} f(i) f(j)$

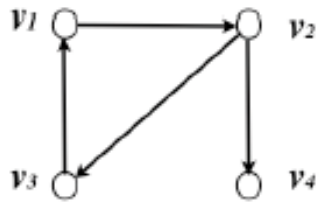


Incidence matrix of a graph

- Dual matrix of adjacency
  - Matrix defined on the edge of the graph

$$\nabla = \begin{cases} \nabla_{ev} = -1, & \text{if } v \text{ is the initial vertex of edge } e \\ \nabla_{ev} = +1, & \text{if } v \text{ is the terminal vertex of edge } e \\ \nabla_{ev} = 0, & \text{if } v \text{ is not in } e \end{cases}$$

$$\nabla = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & +1 \end{bmatrix}$$



  - The mapping $f \to \nabla f$ is known as the co-boundary mapping of *f*

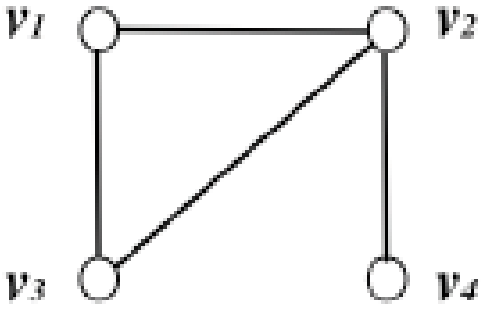
$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & +1 \end{bmatrix} \begin{pmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \end{pmatrix} = \begin{pmatrix} f(2) - f(1) \\ f(1) - f(3) \\ f(3) - f(2) \\ f(4) - f(2) \end{pmatrix}$$

The Laplacian matrix of a graph

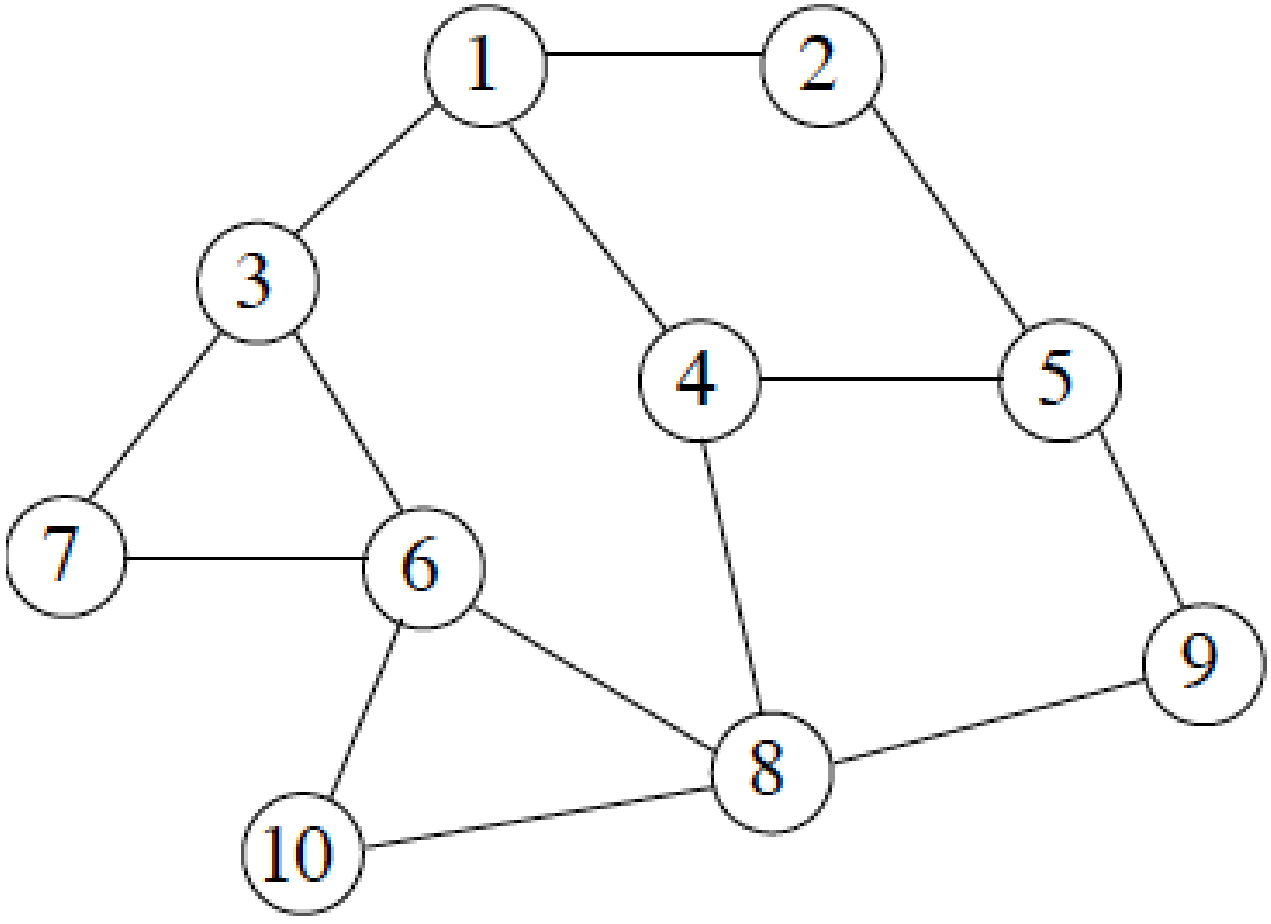
- $L = \nabla^T \nabla$
- $(Lf)(v_i) = \sum_{v_j \to v_i} (f(v_i) - f(v_j))$

- Connection between Laplacian and Adjcency matrix
  - D degree matrix $L = D - W$

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$



A 10 node graph



The adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The laplacian matrix and its eigenvalues

$$\mathbf{L} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 4 & -1 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 2 \end{bmatrix}$$

$$\mathbf{\Lambda} = \begin{bmatrix} 0.0000 & 0.7006 & 1.1306 & 1.8151 & 2.4011 \\ 3.0000 & 3.8327 & 4.1722 & 5.2014 & 5.7462 \end{bmatrix}$$

The Fiedler vector of the graph Laplacian

- The first non-null eigenvalue λ_{k+1} is called the Fiedler value.
  - The corresponding eigenvector is called the Fiedler vector.
- The Fiedler value is the algebraic connectivity of a graph, the further from 0, the more connected.
- The Fiedler vector has been extensively used for spectral bi-partioning

Laplacian regularisation

- Neighborhood graph
  - weighted graph with x_i as vertices $w(x_i, x_j) = h(||x_i - x_j||)$
    - $h(z) = e^{-z^2}$
    - $h(z) = 1_{z \leq t}$
  - Regularizer
$$N(f) = \sum_{i,j}^{n} w(x_i, x_j) \left(f(x_i) - f(x_j)\right)^2$$
    - When $w(x_i, x_j)$ is close to 1 (points are close), $f(x_i) \approx f(x_j)$

Laplacian of a graph

- Laplacian for a weighted graph is defined as $L = D - W$
  - W is the weight matrix, $w_{ij} = w(x_i, x_j)$
  - D is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$
- Laplacian regularization

$$N(f) = f^T L f$$

- Normalized Laplacian $L' = \left(I - D^{-1}W\right)$

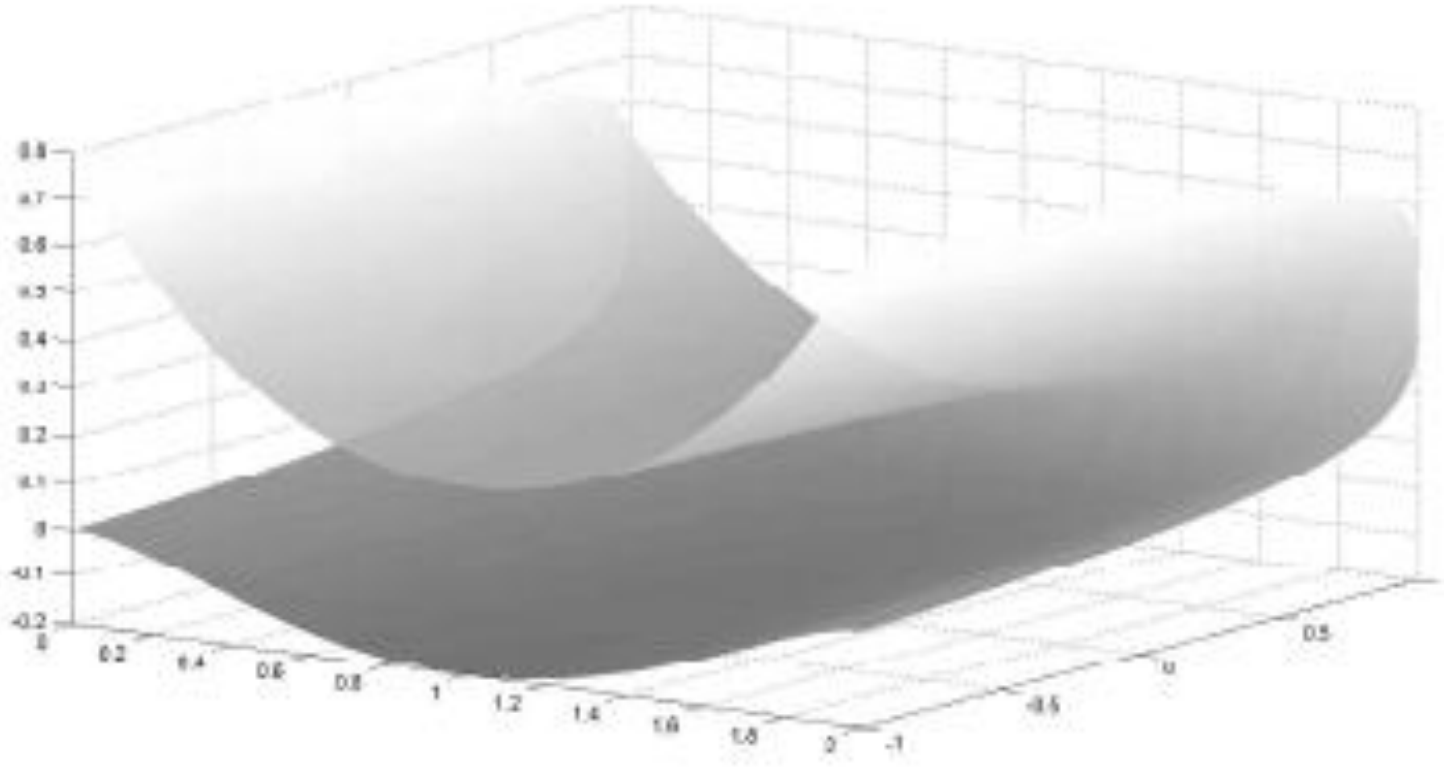
Laplacian embedding

- Embed the graph in a k-dimensional Euclidean space.
  - The embedding is given by the n x k matrix $F = (f_1, \ldots, f_k)$
  - the i-th row of this matrix corresponds to the Euclidean coordinates of the i-th graph node
- The space is obtained by solving

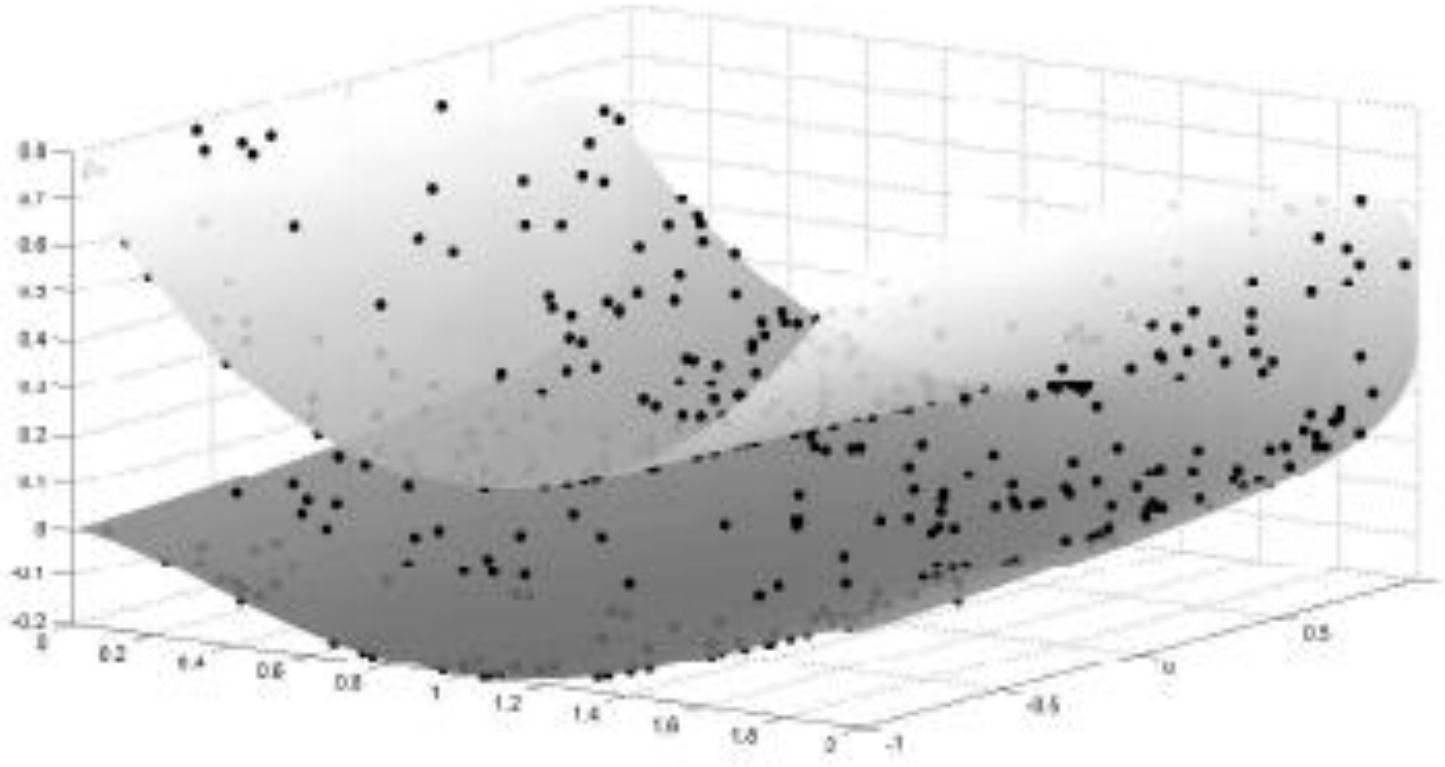
$$\arg \min_{\boldsymbol{f}_1 \cdots \boldsymbol{f}_k} \sum_{i,j=1}^{n} w_{ij} \|\boldsymbol{f}^{(i)} - \boldsymbol{f}^{(j)}\|^2 \text{ with: } \mathbf{F}^\top \mathbf{F} = \mathbf{I}.$$

  - Reduce to finding k lowest non zero eigenvalues of the Laplacian

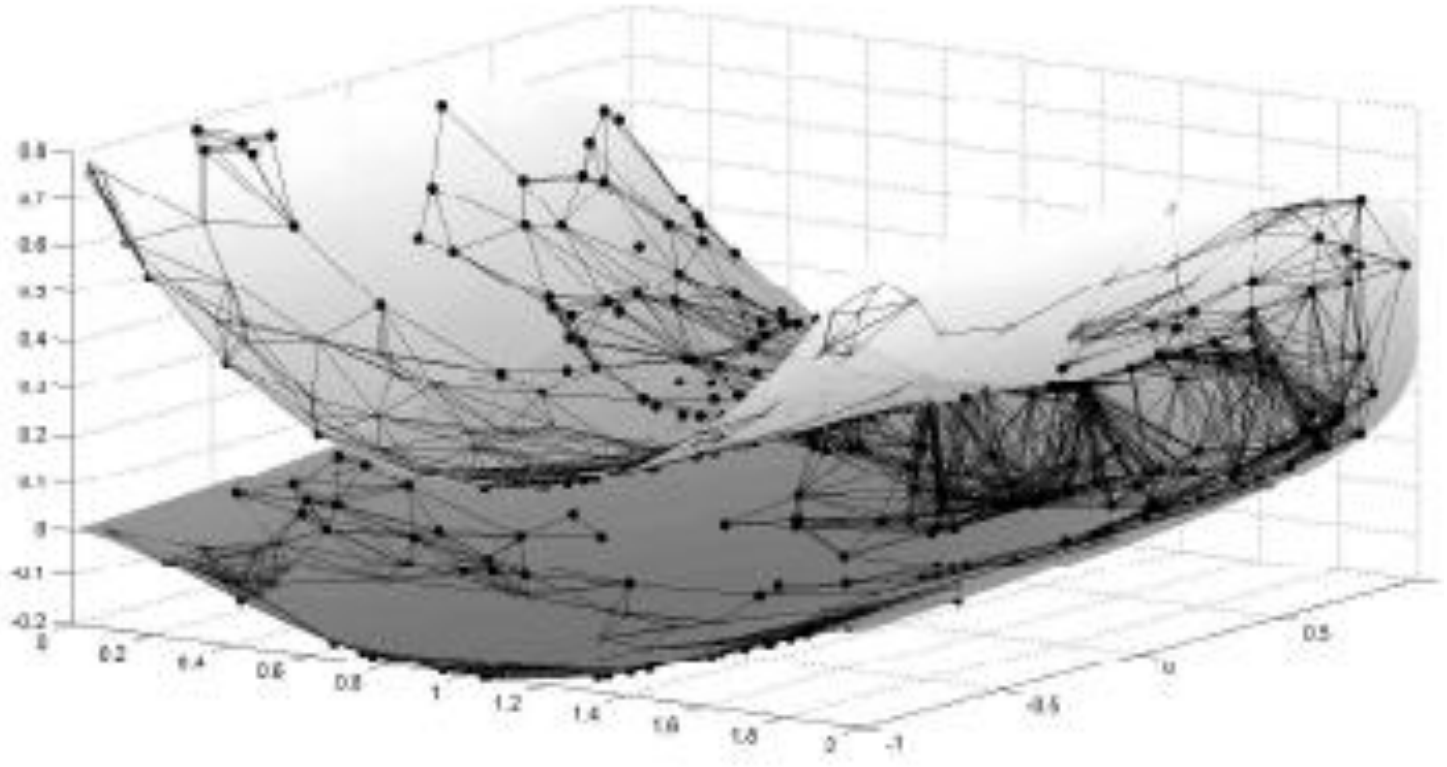
Exemple



Exemple



Exemple



Application

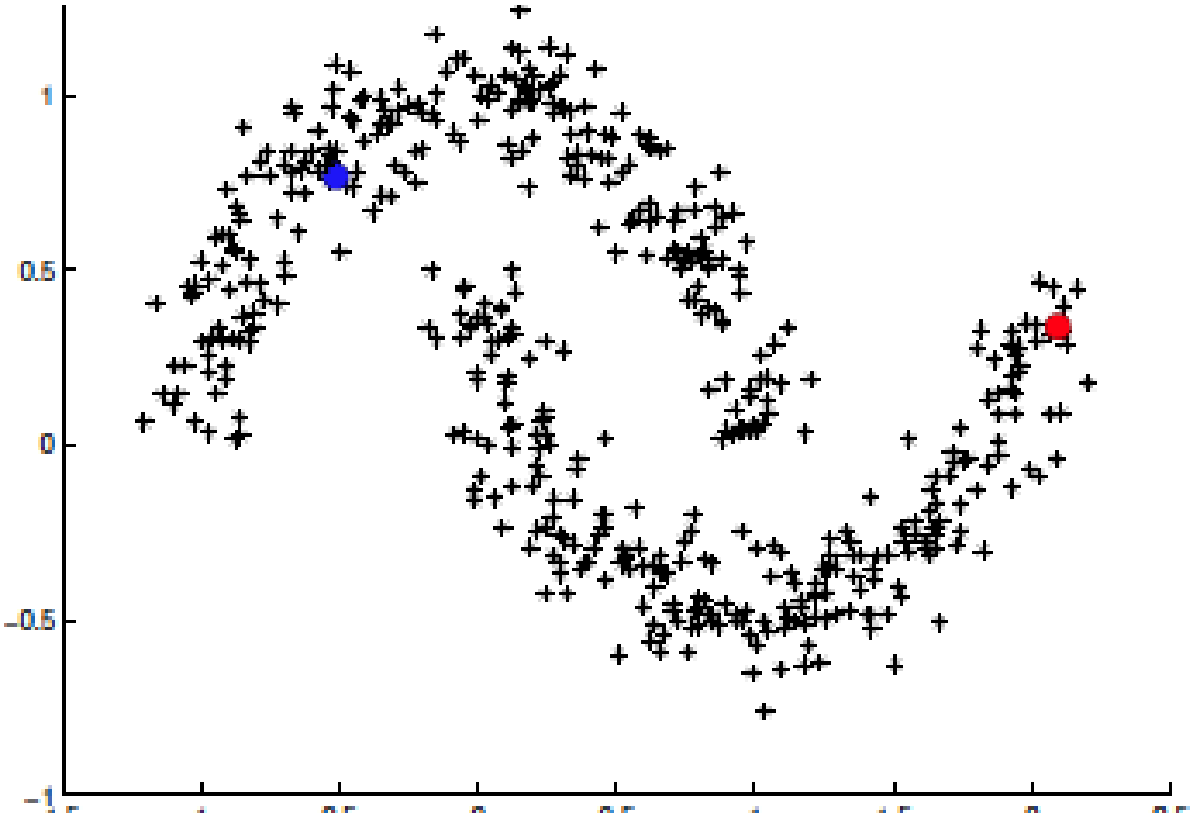
- Dimensionality reduction
  - project on last eigenvectors of L
- Clustering
  - threshold eigenvectors of L, or project first and use k-means afterwards

$$\min_{f \perp 1, ||f||=1} f^T L f$$

- Semi-supervised/supervised
  - use regularization

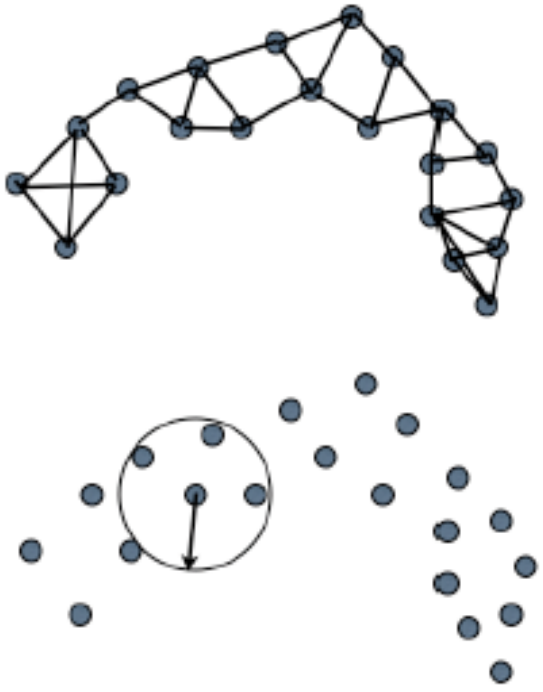
$$\min_f \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda f^T L f$$

Clustering

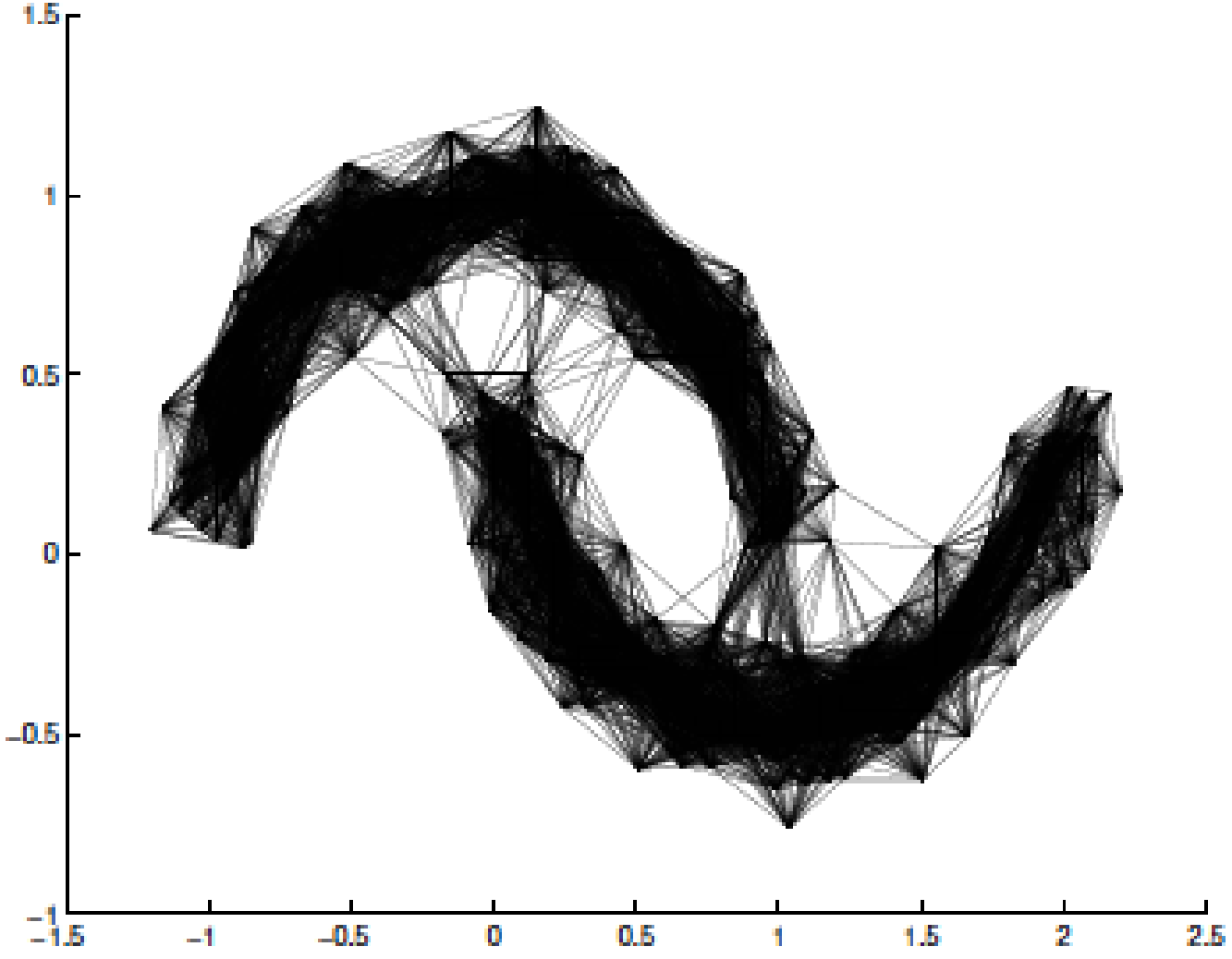


Building a graph from a cloud of points

- K-nearest neighbor
- (KNN) rule
  - ε-radius rule



Graph building



The Graph Partitioning Problem

- We want to find a partition of the graph such that the edges between different groups have very low weight, while the edges within a group have high weight.
  - The mincut problem:
    - Edges between groups have very low weight, and
    - Edges within a group have high weight.
    - Choose a partition of the graph into k groups that minimizes the following criterion: $\text{mincut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i)$

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

Ratio Cut and Normalized Cut

- Often, the mincut solution isolates a vertex from the rest of the graph.
  - Request that the groups are reasonably large.
- Ratio cut minimizes:

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|}$$

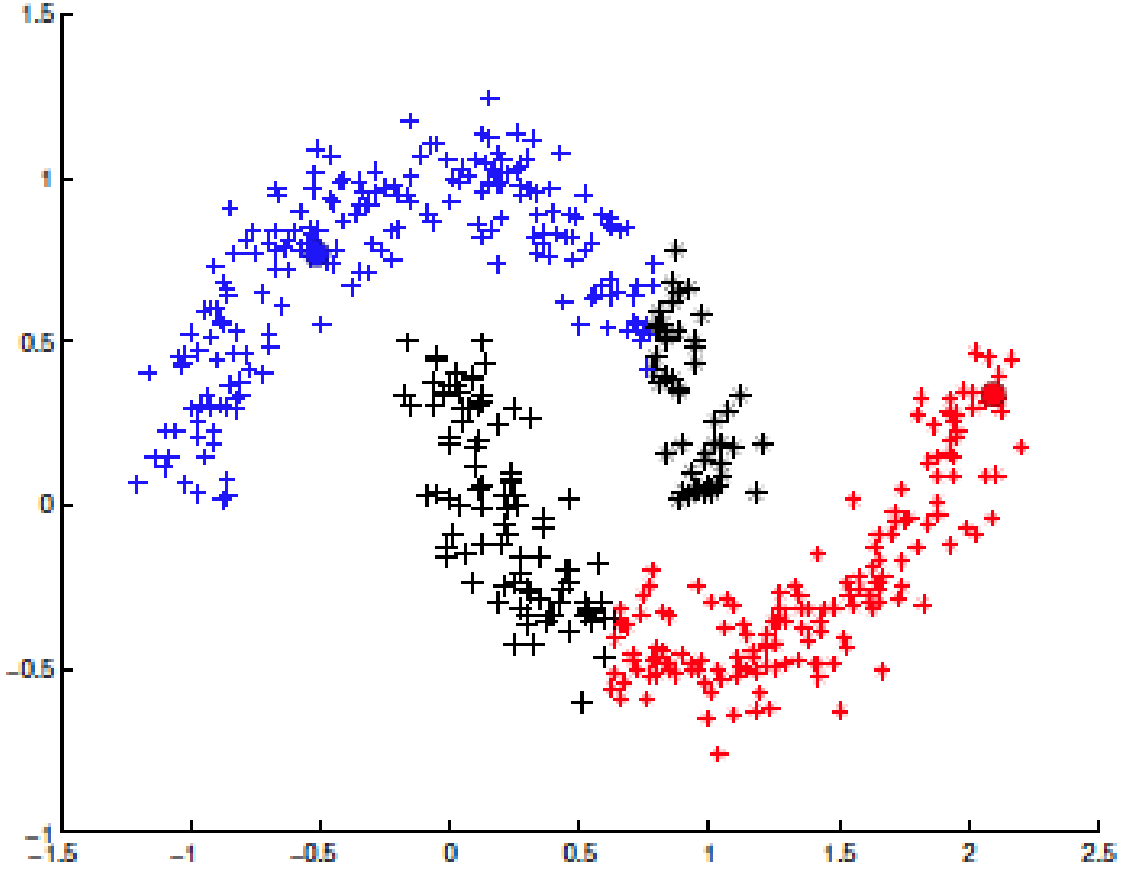
- Normalized cut:

$$\text{NCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{\kappa} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)}$$

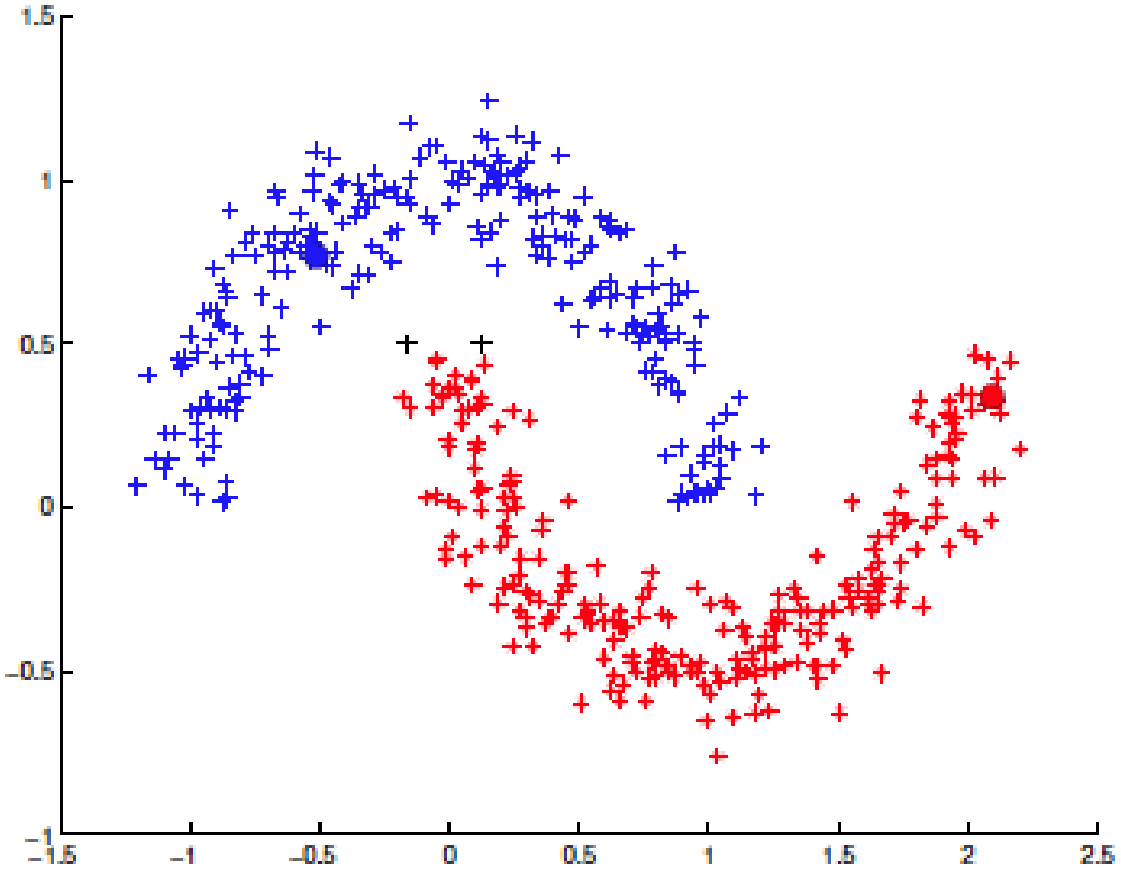
What is Spectral Clustering?

- Both ratio-cut and normalized-cut minimizations are NP-hard problems
  - Spectral clustering is a way to solve relaxed versions of these problems:
- The smallest non-null eigenvectors of the unnormalized Laplacian approximate the RatioCut minimization criterion,
- The smallest non-null eigenvectors of the random-walk Laplacian approximate the NormalizedCut criterion.

Application (clustering)



Application (with regularization)



Why it works ?

- Two cases:
  - the neighborhood size is fixed (von Luxburg, B., Belkin 2005)
  - the neighborhood size goes to zero as n increases (Hein, Audibert 2005, 2006)

- On a compact manifold $\mathcal{M}$ with metric g and density p (wrt μ), under technical conditions :

$$\lim_{n\to\infty} f^T L f = \int_{\mathcal{M}} ||\nabla f||^2_{\mathcal{M}} p^2 \sqrt{\det g} d\mu$$

Conclusion

- Goal is not to identify the manifold, but to exploit the (approximate) low-dimensionality / clustered-ness of the data
  - Transpose manifolds to graphs (finite set of data)
  - Very active area of research (best semi-supervised algorithms use this idea)
  - Theory very limited
- Many algorithmic issues (choice of the graph, weights, regularizer...)
  - Large application potential