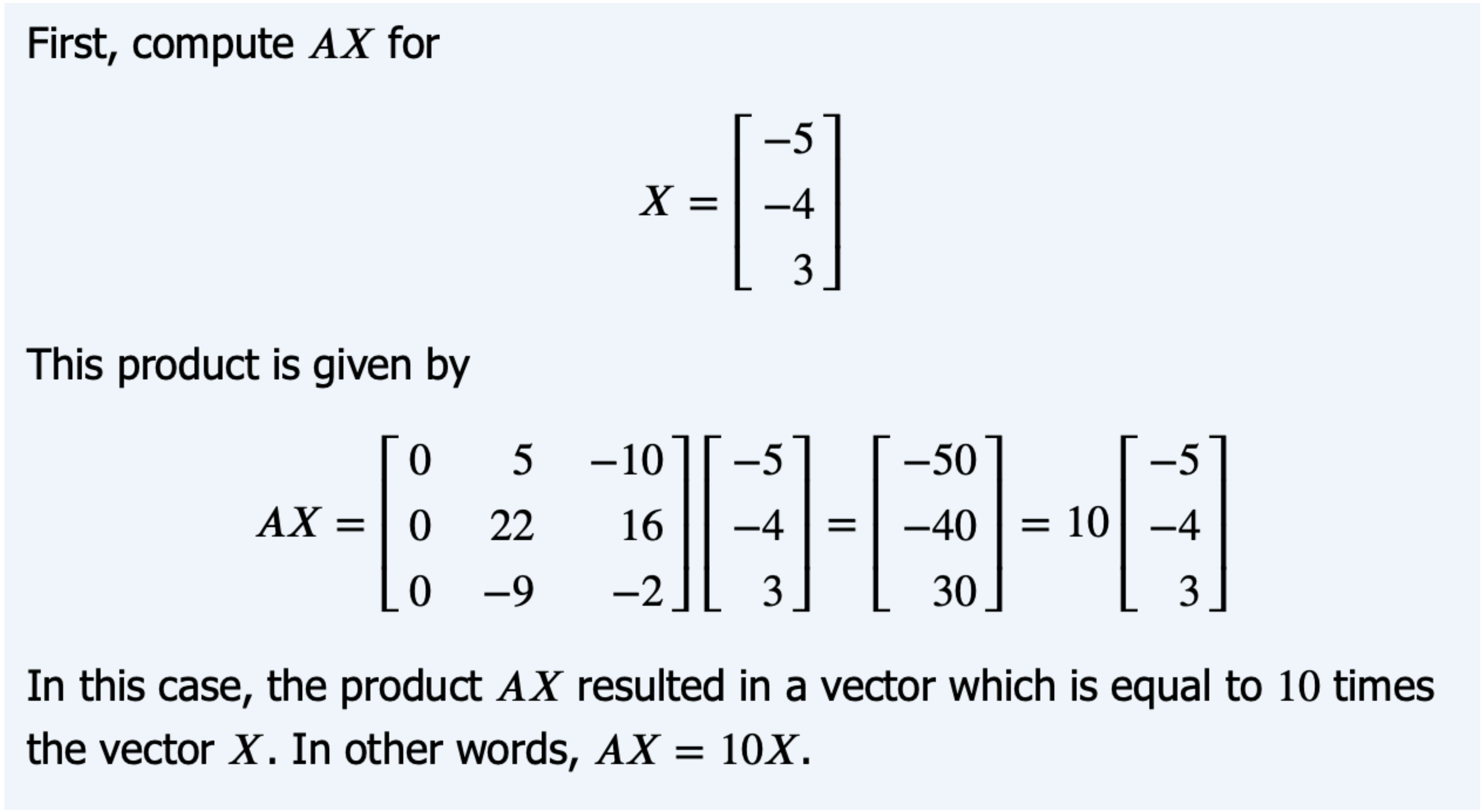
Spectral clustering

Reading https://people.csail.mit.edu/dsontag/courses/ml14/notes/Luxburg07_tutorial_spectral_clustering.pdf

Spectral clustering

Steps

- Preprocess the data
- Find the graph Laplacian vector of the matrix
- Use K-means on eigenvectors of the graph Laplacian to cluster



First, compute AX for

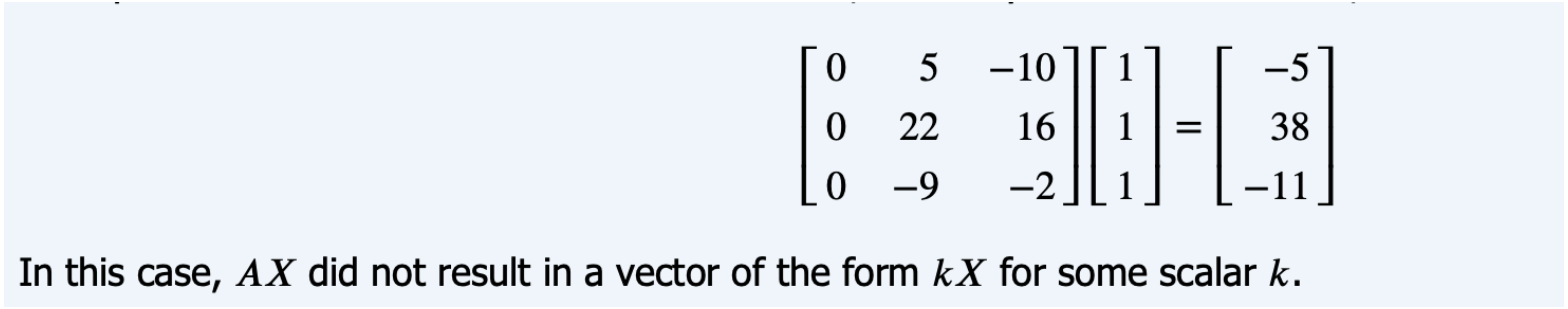
$$X = \begin{bmatrix} -5 \\ -4 \\ 3 \end{bmatrix}$$

This product is given by

$$AX = \begin{bmatrix} 0 & 5 & -10 \\ 0 & 22 & 16 \\ 0 & -9 & -2 \end{bmatrix} \begin{bmatrix} -5 \\ -4 \\ 3 \end{bmatrix} = \begin{bmatrix} -50 \\ -40 \\ 30 \end{bmatrix} = 10 \begin{bmatrix} -5 \\ -4 \\ 3 \end{bmatrix}$$

In this case, the product AX resulted in a vector which is equal to 10 times the vector X. In other words, $AX = 10X$.

Perhaps this is a special matrix A which when multiplied by any vector $\mathbf{x}$ always equals $k\mathbf{x}$?



$$\begin{bmatrix} 0 & 5 & -10 \\ 0 & 22 & 16 \\ 0 & -9 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -5 \\ 38 \\ -11 \end{bmatrix}$$

In this case, AX did not result in a vector of the form kX for some scalar k.

NOPE!

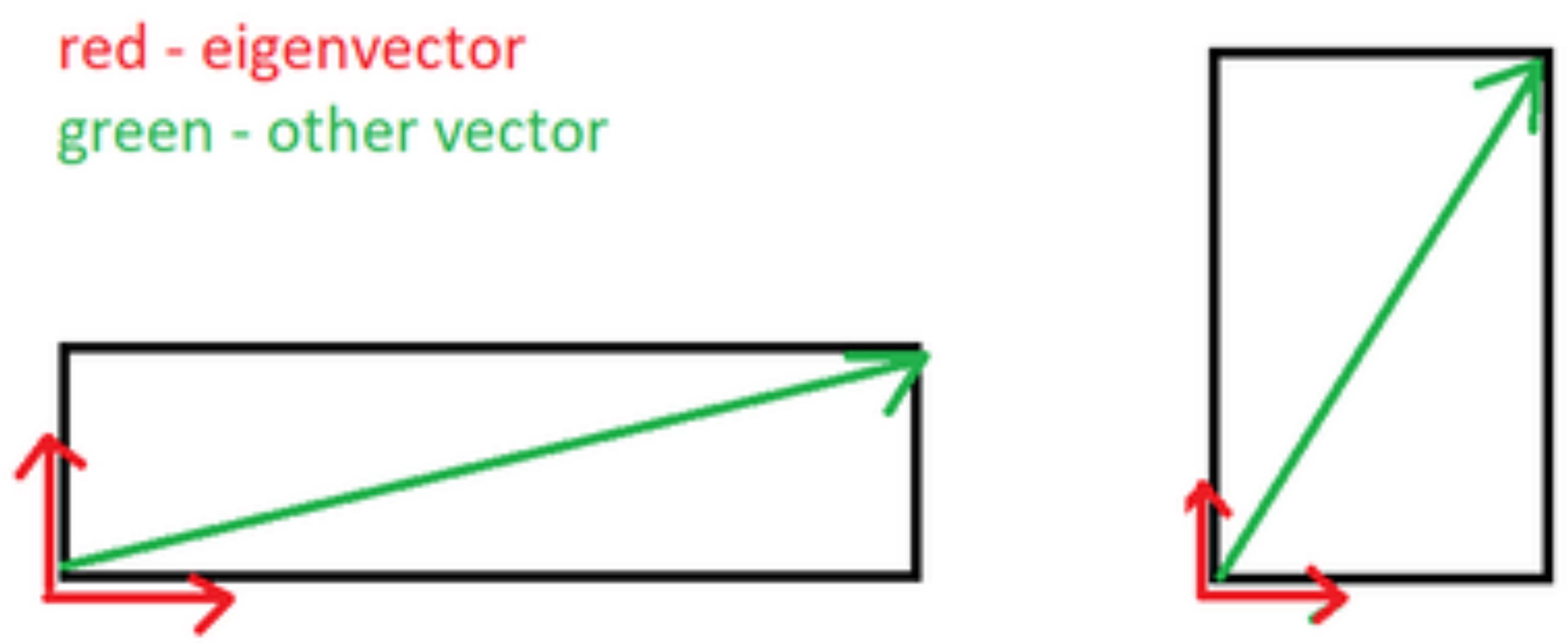
https://math.libretexts.org/Bookshelves/Linear_Algebra/A_First_Course_in_Linear_Algebra_(Kuttler)/07:_Spectral_Theory/7.01:_Eigenvalues_and_Eigenvectors_of_a_Matrix

Eigenvectors and Eigenvalues

When $A \in \mathbb{R}^{n \times n}$ is positive semidefinite then a vector $\mathbf{v}$ is an eigenvector and has associated eigenvalue λ iff

$$(A - \lambda \mathrm{I})\mathbf{v} = 0$$
$$A\mathbf{v} = \lambda \mathbf{v}$$

That is eigenvectors do not change direction, they only scale linearly when A is subject to a linear transformation.



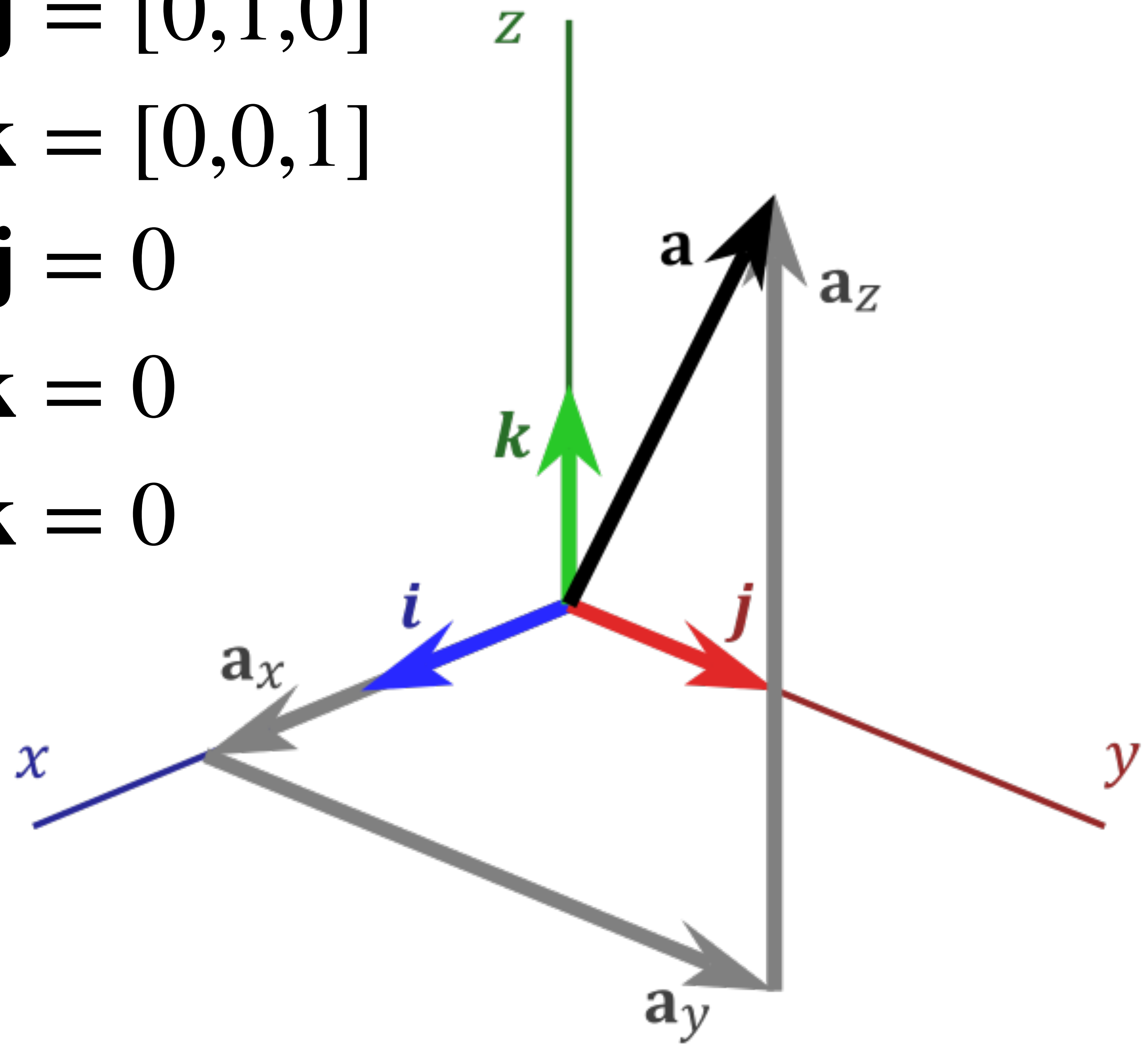
Eigenvectors and Eigenvalues

If $A \in \mathbb{R}^{n\times n}$ is symmetric then there exists an orthogonal set of eigenvectors.

Define $U = [\mathbf{v_1}, \mathbf{v_2}\ldots\mathbf{v_n}]$, a square matrix whose columns are the n linearly independent eigenvectors of $\mathbf{A}$. Also define the vector of matching eigenvalues $\sigma(A) = [\lambda_1, \lambda_2\ldots\lambda_n]$; this vector is known as the spectrum of A. Lastly define Λ, the diagonalization of the spectrum $\Lambda = \mathrm{I}\sigma(A)$

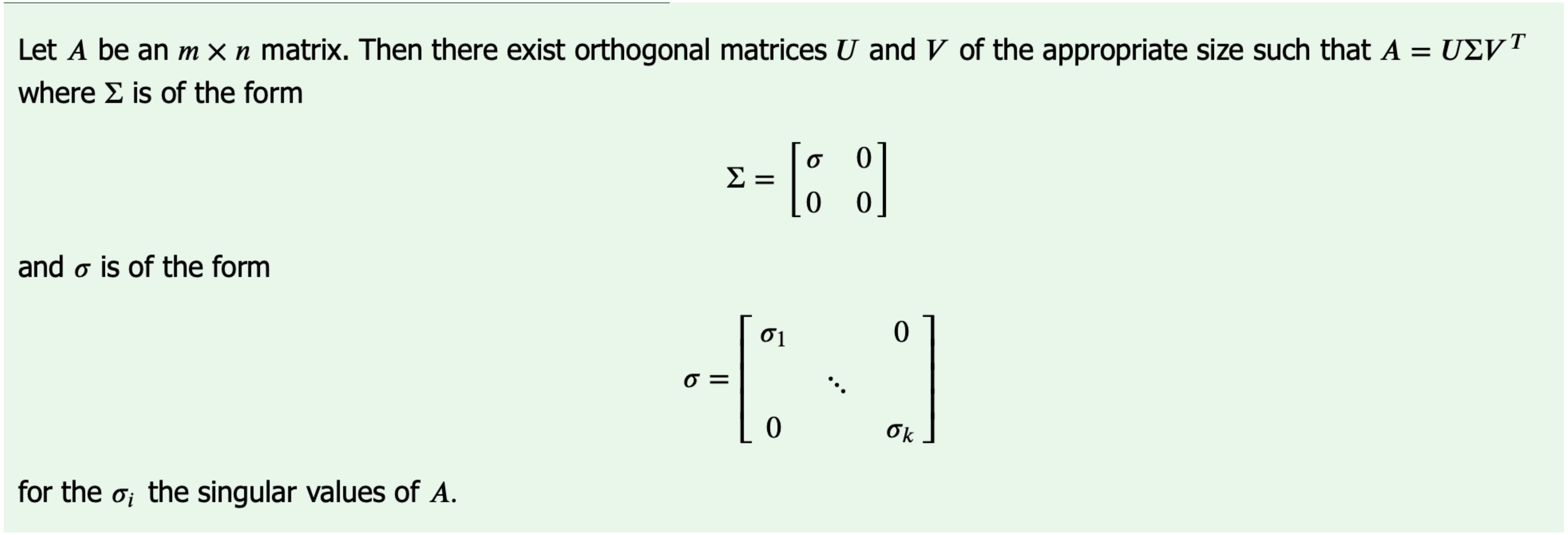
It can be shown that $U^{-1}AU = \Lambda$

$$\mathbf{i} = [1{,}0{,}0]$$
$$\mathbf{j} = [0{,}1{,}0]$$
$$\mathbf{k} = [0{,}0{,}1]$$
$$\mathbf{i}\cdot\mathbf{j} = 0$$
$$\mathbf{j}\cdot\mathbf{k} = 0$$
$$\mathbf{i}\cdot\mathbf{k} = 0$$



Eigenvectors and Eigenvalues

Can be generalized to $A \in \mathbb{R}^{n \times m}$ if A is invertible (full rank/positive definite) through Singular Value Decomposition



Let A be an $m \times n$ matrix. Then there exist orthogonal matrices U and V of the appropriate size such that $A = U\Sigma V^T$ where Σ is of the form

$$\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & 0 \end{bmatrix}$$

and σ is of the form

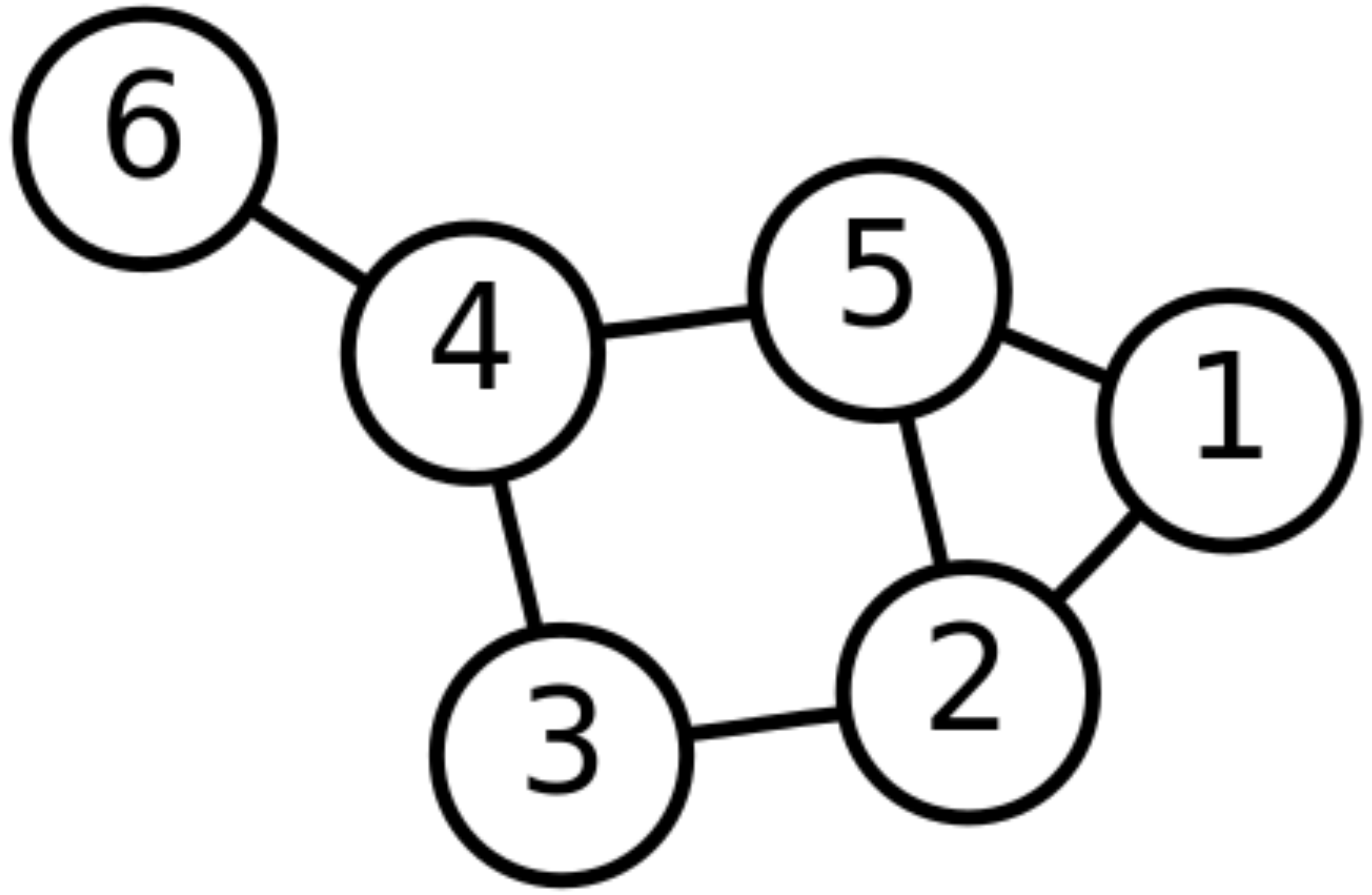
$$\sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{bmatrix}$$

for the σ_i the singular values of A.

In one restricted but very common sense of the term,[1][2] a **graph** is an ordered pair $G = (V, E)$ comprising:

- V, a set of **vertices** (also called **nodes** or **points**);
- $E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\}$, a set of **edges** (also called **links** or **lines**), which are unordered pairs of vertices (that is, an edge is associated with two distinct vertices).

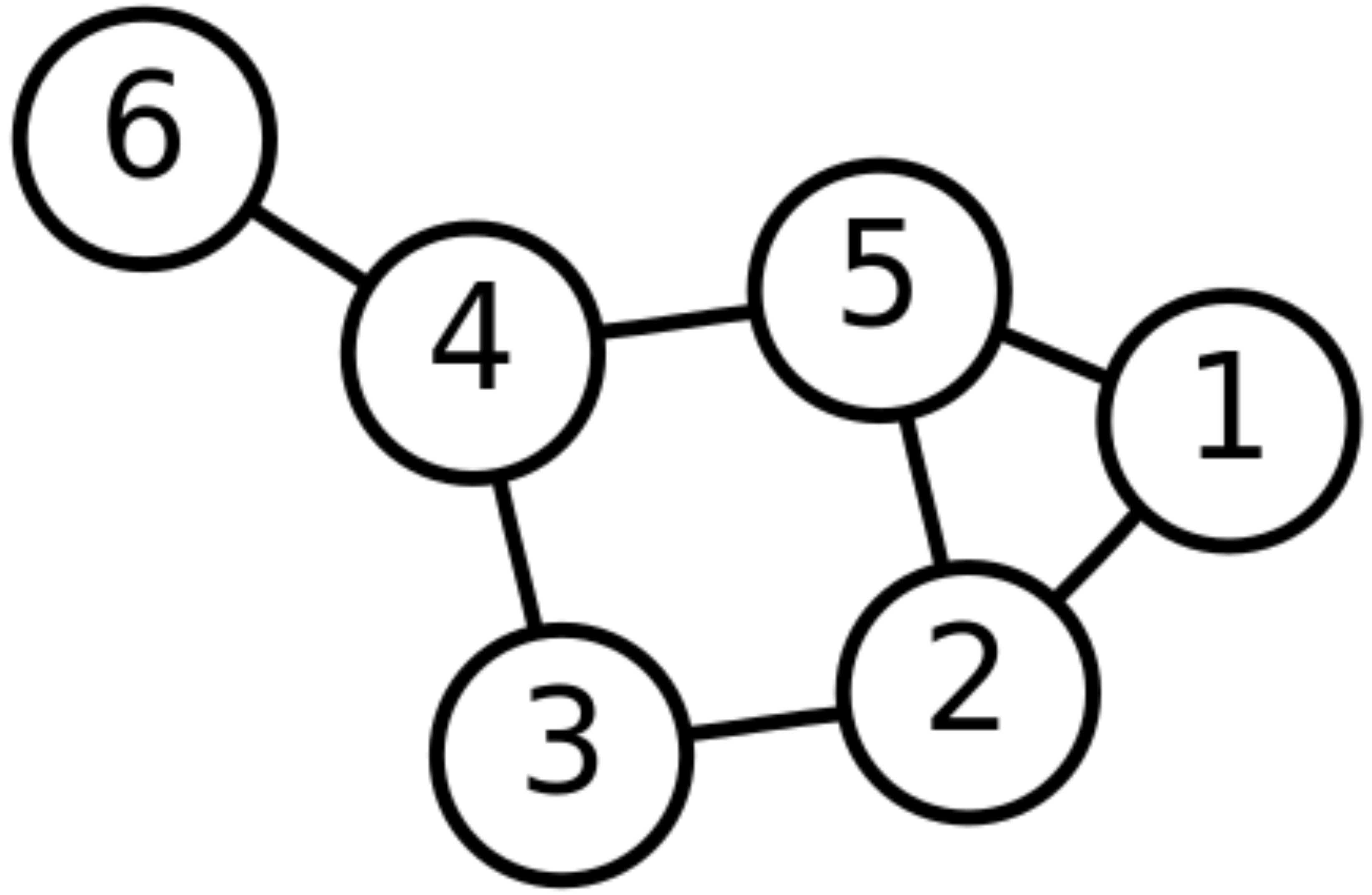
To avoid ambiguity, this type of object may be called precisely an **undirected simple graph**.



Similarity graph

There are many ways to make a graph from data

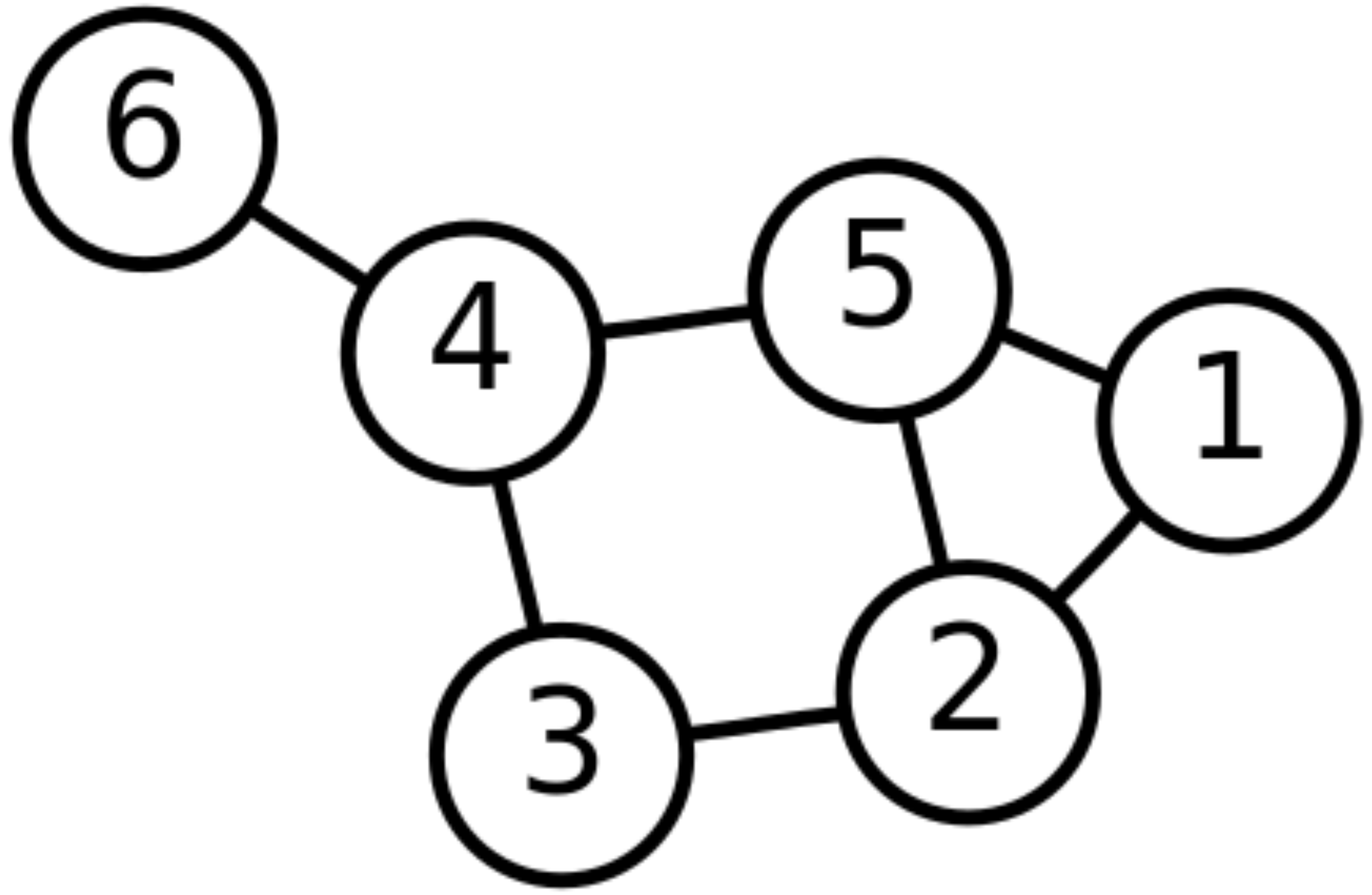
- Unweighted
  - $W_{i,j} = 1$ iff $d(x_i, x_j) < \epsilon \; \forall i, j \neq i \in G$; otherwise 0
  - $W_{i,j} = 1$ iff x_i is in the set of k nearest neighors of x_j; otherwise 0
- Weighted
  - $W_{i,j} = \exp(\frac{\|x_i - x_j\|^2}{2\sigma^2})$



Affinity matrix

Unweighted example

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

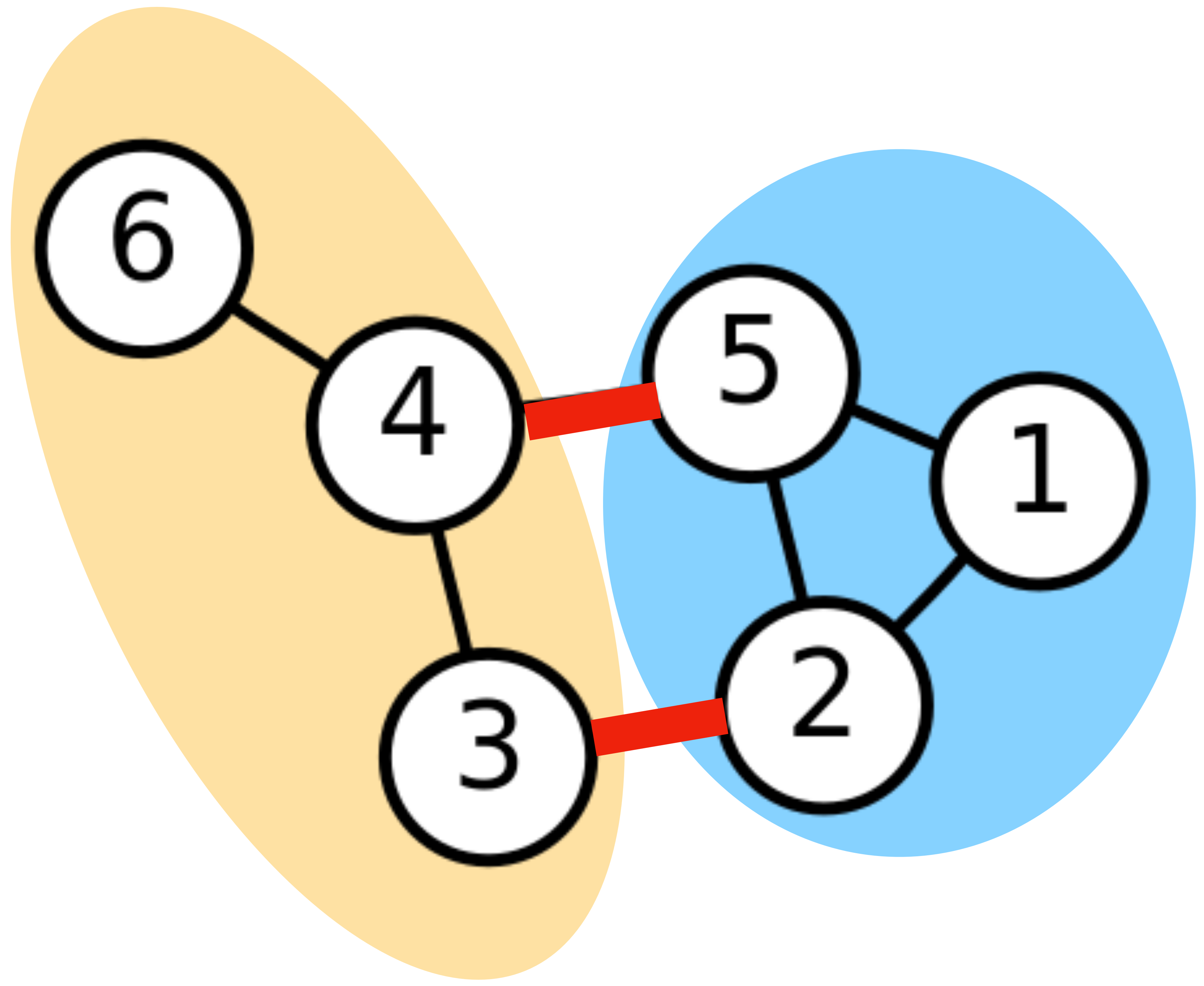


Clustering as a bi-partitioning task

Partition vertices into two disjoint groups A, B

How can we define a “good” partition?

How can we efficiently calculate such a partition?



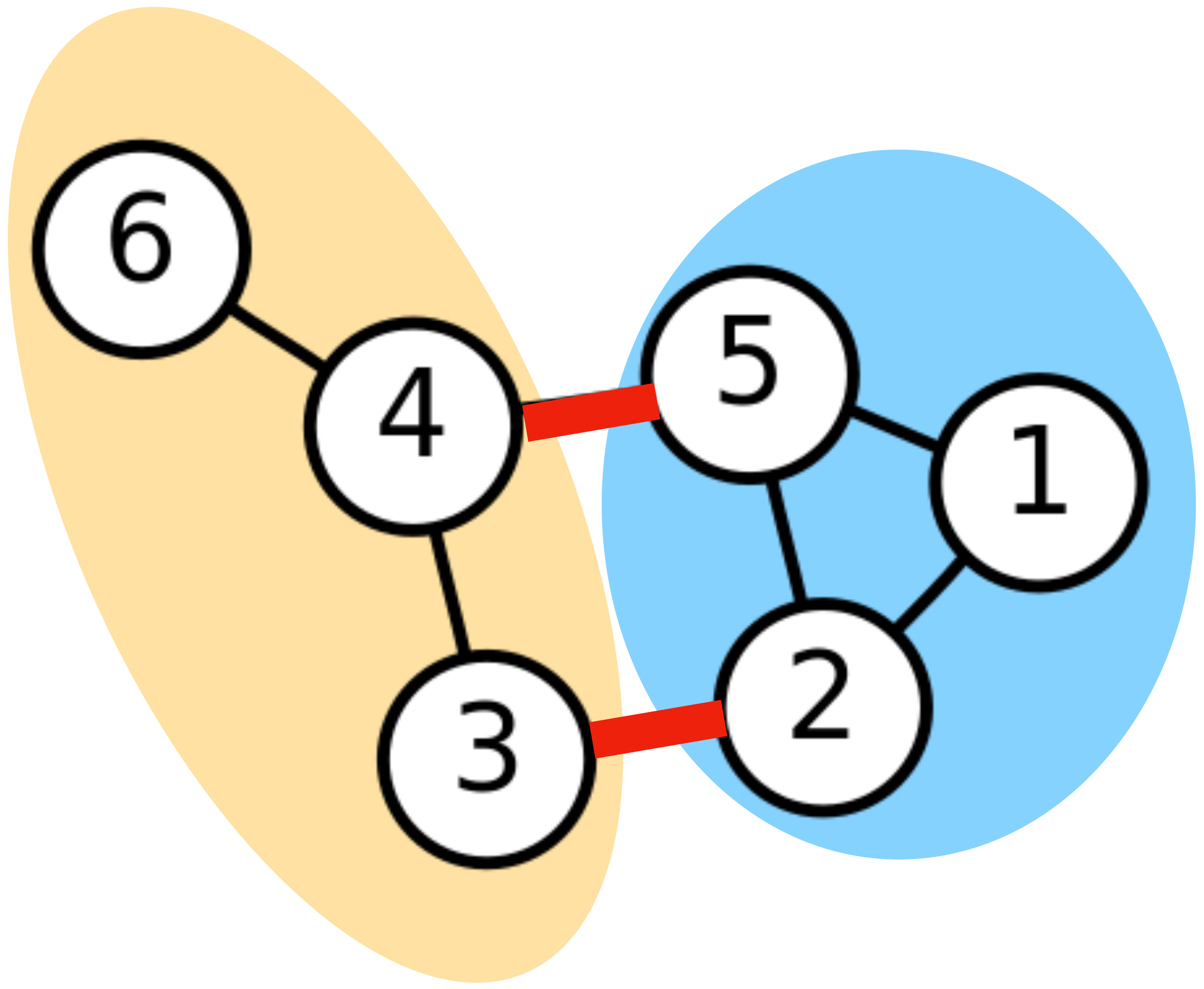
Clustering as a bi-partitioning task

Partition vertices into two disjoint groups A, B

$$\text{cut(A,B)} = \sum_{\forall i \in A, \forall j \in B} W_{i,j}$$

And just

$$\arg\min_{A,B} \text{cut(A,B)}?$$

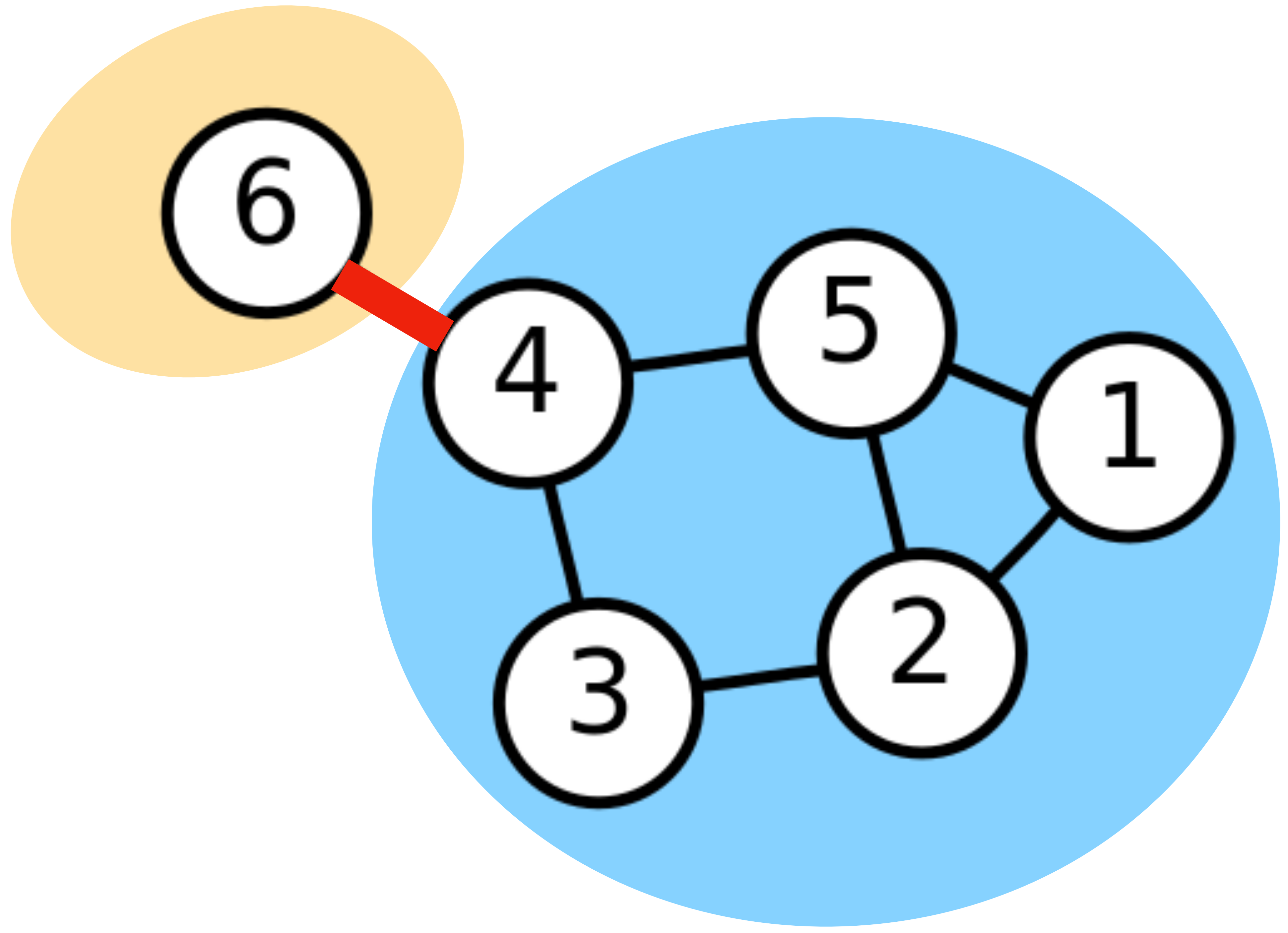


Clustering as a bi-partitioning task

Partition vertices into two disjoint groups A, B

$$\arg\min_{A,B} \text{cut}(A,B)$$

Degenerate case!



Clustering as a bi-partitioning task

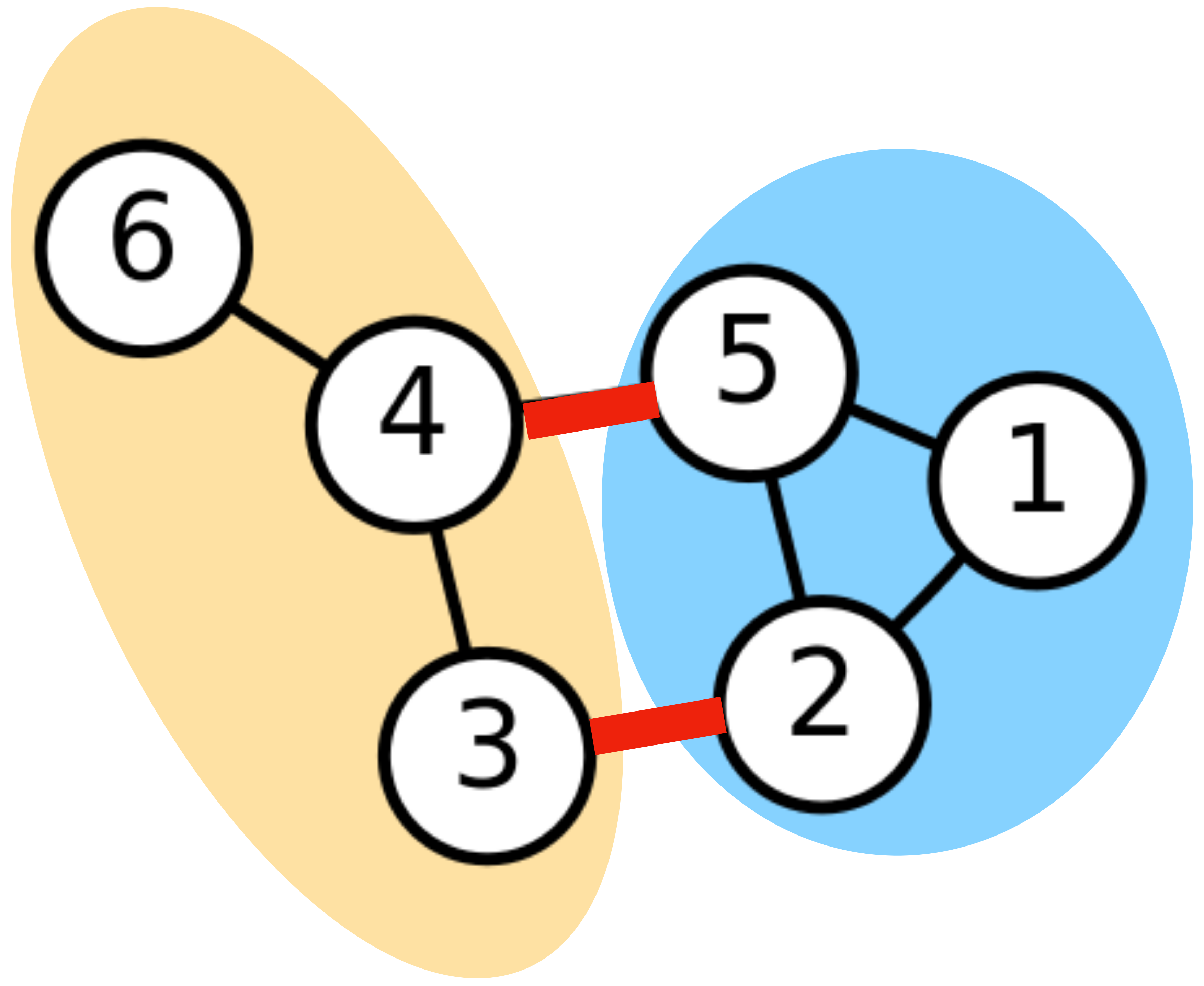
Partition vertices into two disjoint groups A, B

Normalized cut forces clusters to have larger size

$$\text{Ncut(A,B)} = \text{cut(A,B)} \left(\frac{1}{\text{Vol}(A)} + \frac{1}{\text{Vol}(B)}\right)$$

Where Vol(A) is the sum of the degrees of all of the nodes in A

Then we just $\arg\min_{A,B}$ Ncut(A,B)



Clustering as a bi-partitioning task

Partition vertices into two disjoint groups A, B

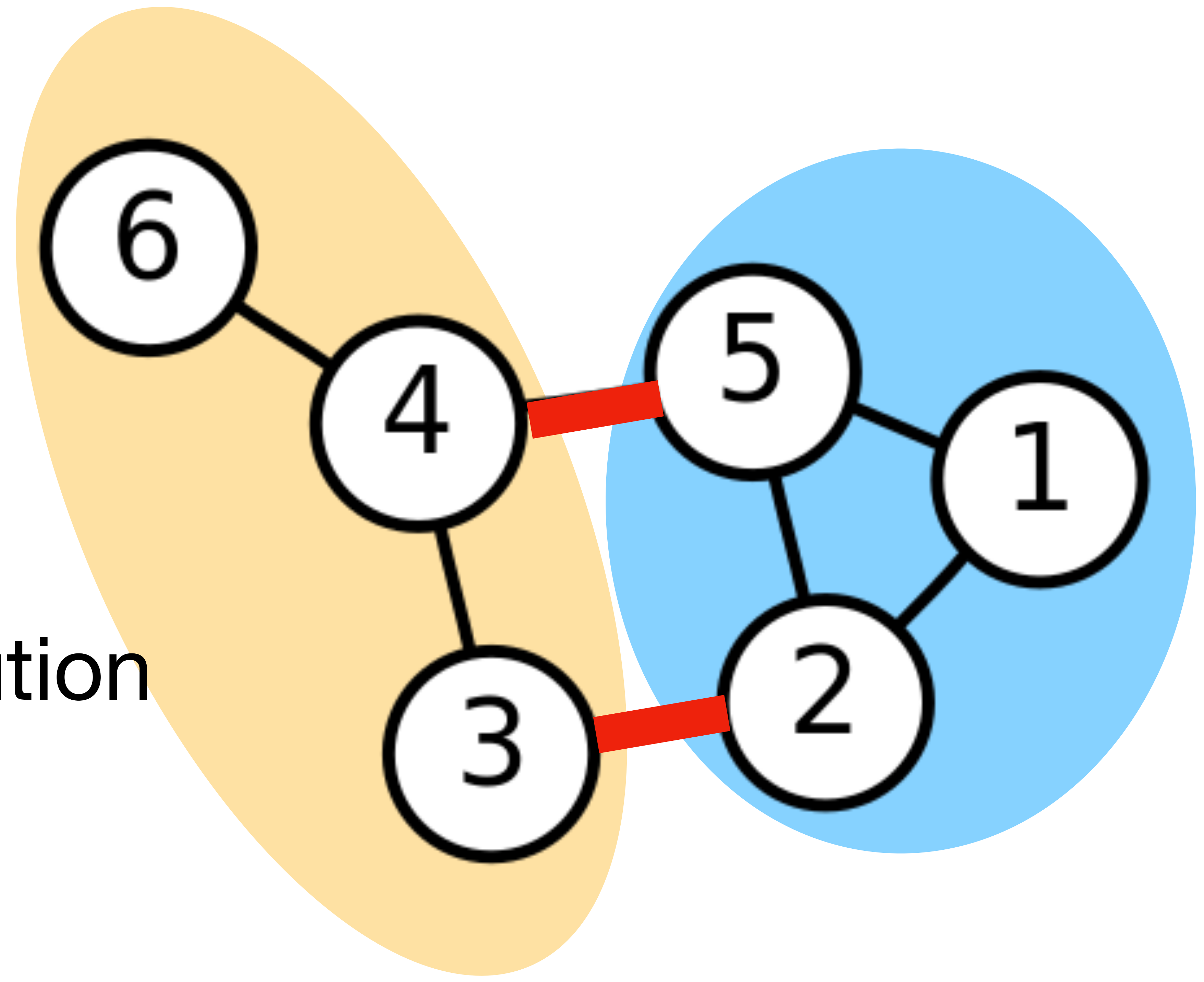
How can we define a “good” partition?

OK!

How can we efficiently calculate such a partition?

Not this way… NP-Hard… lets make a “relaxed” solution

Involving eigenvectors of the graph Laplacian matrix



Laplacian matrix [edit]

Given a simple graph G with n vertices $v_1, \ldots, v_n$, its Laplacian matrix $L_{n\times n}$ is defined element-wise as[1]

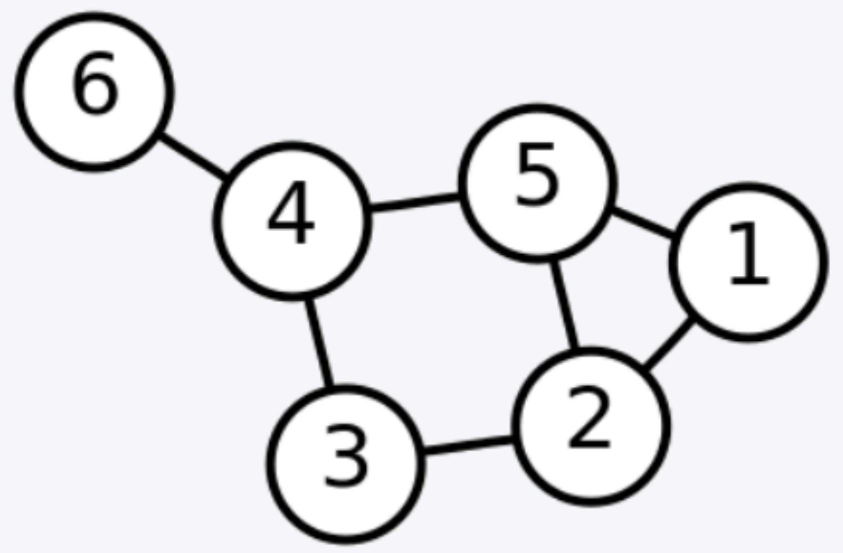
$$L_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise,} \end{cases}$$

or equivalently by the matrix

$$L = D - A,$$

where *D* is the degree matrix and *A* is the adjacency matrix of the graph. Since G is a simple graph, A only contains 1s or 0s and its diagonal elements are all 0s.

Here is a simple example of a labelled, undirected graph and its Laplacian matrix.

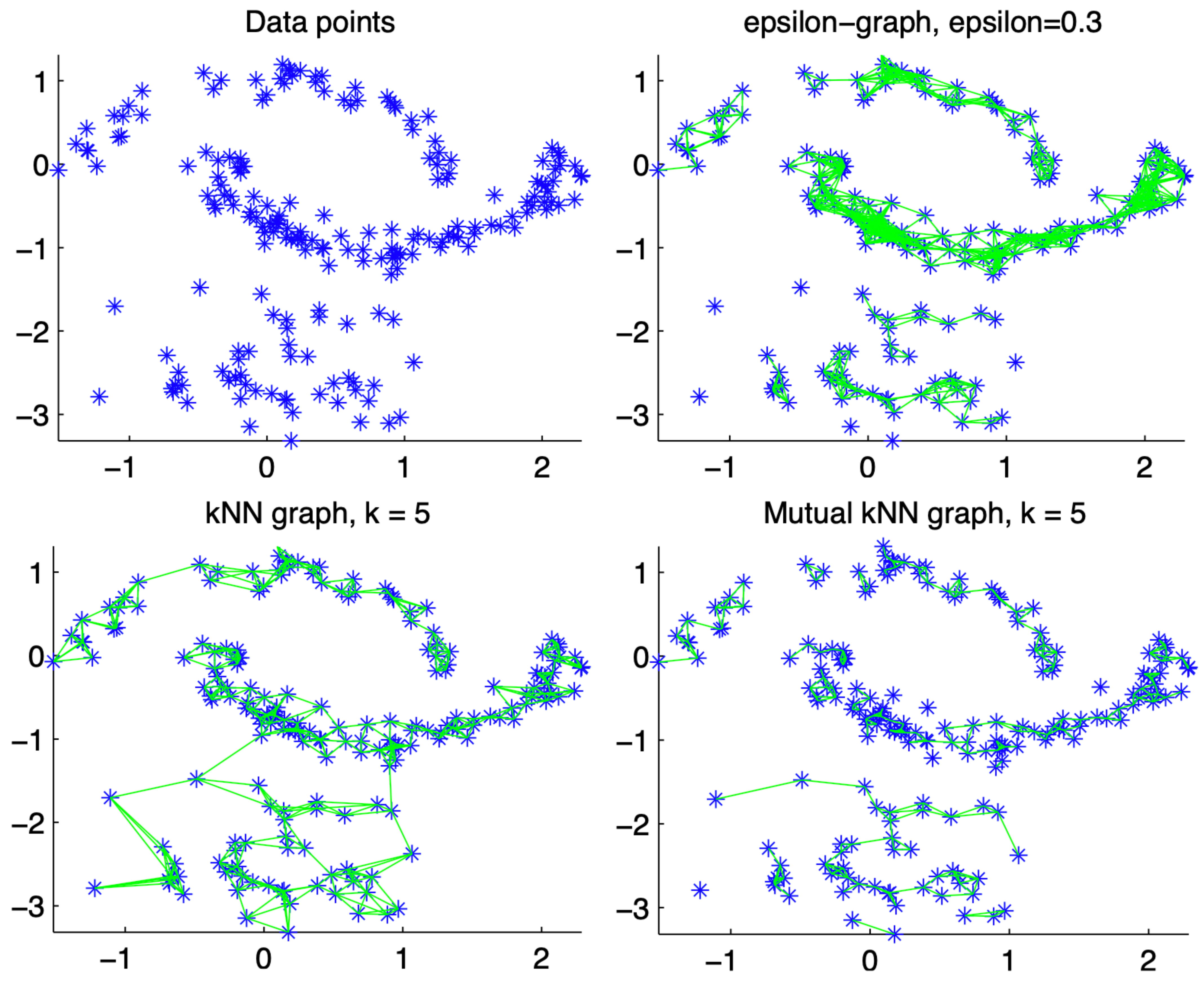
Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$

We observe for the undirected graph that both the adjacency matrix and the Laplacian matrix are symmetric, and that row- and column-sums of the Laplacian matrix are all zeros.

Lots of different similarity graphs and graph Laplacians!

$$L_{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{\text{rw}} := D^{-1} L = I - D^{-1} W.$$



Normalized Cut and Graph Laplacian

$$\mathrm{Ncut}(A, B) := \mathrm{cut}(A, B)\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)$$

Let f = $[f_1\ f_2\ \dots\ f_n]^T$ with $f_i = \begin{cases} \frac{1}{\mathrm{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)} & \text{if } i \in B \end{cases}$

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{ij} w_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2 = \sum_{i\in A, j\in B} w_{ij}\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)^2$$

$$\mathbf{f}^T\mathbf{D}\mathbf{f} = \sum_{j} d_i \mathbf{f}_i^2 = \sum_{i\in A} \frac{d_i}{\mathrm{vol}(A)^2} + \sum_{j\in B} \frac{d_i}{\mathrm{vol}(B)^2} = \frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}$$

$$\mathrm{Ncut}(A, B) = \frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$$

Normalized Cut and Graph Laplacian

$$\min \text{Ncut}(A, B) = \min \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$$

where $f = [f_1\ f_2\ \dots\ f_n]^T$ with $f_i = \begin{cases} \frac{1}{\text{vol}(A)} & \text{if } i \in A \\ -\frac{1}{\text{vol}(B)} & \text{if } i \in B \end{cases}$

Relaxation: $\min \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$ s.t. $f^T D 1 = 0$

Solution: f – second eigenvector of generalized eval problem

$$\boxed{\mathbf{L}\mathbf{f} = \lambda \mathbf{D}\mathbf{f}}$$

Obtain cluster assignments by thresholding f at 0

Approximation of Normalized cut

$$\mathrm{Ncut}(A,B) := \mathrm{cut}(A,B)\left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}\right)$$

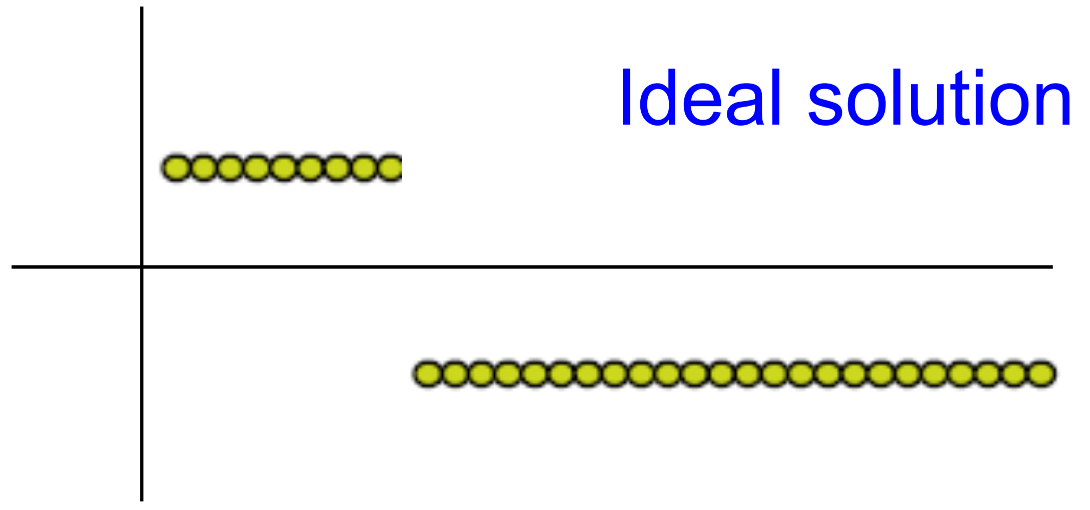
Let *f* be the eigenvector corresponding to the second smallest eval of the generalized eval problem.

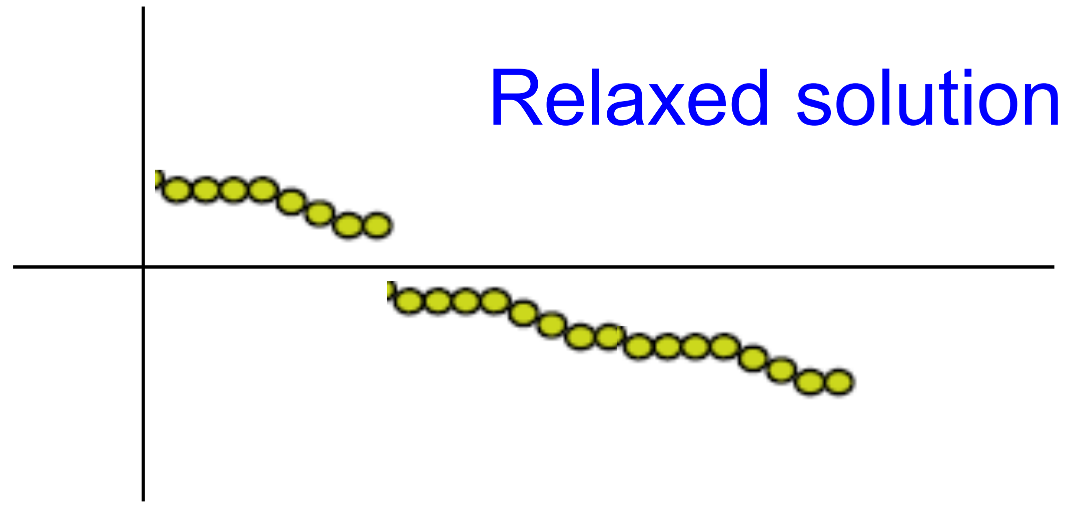
$$\boxed{\mathrm{Lf} = \lambda \mathrm{Df}}$$

Equivalent to eigenvector corresponding to the second smallest eval of the normalized Laplacian $L' = D^{-1}L = I - D^{-1}W$

Recover binary partition as follows:

$i \in A$	*if*	$f_i \geq 0$
$i \in B$	*if*	$f_i < 0$





Spectral clustering

Steps

- Preprocess the data
    - Create a matrix describing connectivity between datapoints (e.g. kNN) or similarity between datapoints (Gaussian kernel)
- Find the graph Laplacian vector of the matrix
    - Several different Laplacians! Each have different properties! Not just one spectral clustering
- Use K-means on evecs/evals of the graph Laplacian to cluster
    - Multiplicity of 0 eigenvalues are the number of connected components in the graph. So if there's no overlap between clusters)… then these smaller evecs assign datapoints to clusters
    - Other options
        - If binary clustering: the 1st (smallest eigenvalue) eigenvector is al 1s, then the sign of the 2nd eigenvector (so-called Fiedler eigenvector) components are the cluster assignments
        - If k clusters: take eigenvectors 2 through k+1 as indicator variables