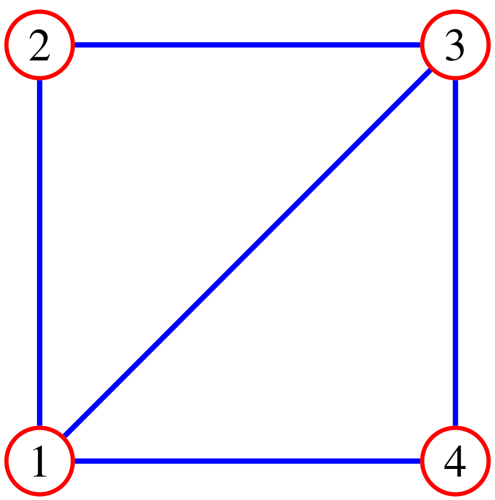
7. Spectral clustering

- Laplacian matrix
- graph partitioning
- spectral clustering

Undirected graph

$$G = (V, E)$$

- V is a finite set of *vertices*; we will assume $V = \{1, 2, \ldots, n\}$
- $E \subseteq \{\{i, j\} \mid i, j \in V\}$ is the set of (undirected) *edges*
- two vertices i and j are *adjacent* if $\{i, j\} \in E$
- the *neighborhood* $\mathcal{N}(i)$ of vertex i is the set of vertices adjacent to i



$$\begin{aligned} V &= \{1, 2, 3, 4\} \\ E &= \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{3, 4\}\} \\ \mathcal{N}(4) &= \{1, 3\} \end{aligned}$$

Edge weights

Weights: each edge $\{i, j\}$ has a positive weight $W_{ij} = W_{ji}$

- if all the edge weights are one the graph is called *unweighted*
- we define $W_{ij} = 0$ if i and j are not adjacent ($\{i, j\}$ is not an edge) or if $i = j$
- the symmetric matrix W with elements W_{ij} is the (weighted) *adjacency matrix*

edge weights express strength of connection, association, similarity of vertices

Degree: the *degree* of a vertex is the sum of the weights of the incident edges

$$\deg(i) = \sum_{j \in \mathcal{N}(i)} W_{ij} = \sum_{j=1}^{n} W_{ij} = (W\mathbf{1})_i$$

in the example on the previous page, $\deg(4) = W_{14} + W_{34}$

Graph Laplacian

Graph Laplacian: the symmetric $n \times n$ matrix

$$
\begin{aligned}
L &= \mathbf{diag}(W\mathbf{1}) - W \\
&= \begin{bmatrix} \deg(1) & -W_{12} & \cdots & -W_{1n} \\ -W_{21} & \deg(2) & \cdots & -W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -W_{n1} & -W_{n2} & \cdots & \deg(n) \end{bmatrix}
\end{aligned}
$$

Normalized graph Laplacian: includes a symmetric scaling of rows and columns

$$L_{\mathrm{n}} = \mathbf{diag}(W\mathbf{1})^{-1/2} L \, \mathbf{diag}(W\mathbf{1})^{-1/2}$$

normalized Laplacian has unit diagonal, off-diagonal elements

$$(L_{\mathrm{n}})_{ij} = \frac{-W_{ij}}{\sqrt{\deg(i)\deg(j)}}$$

Laplacian as Gram matrix

the Laplacian can be written as a Gram matrix (page 2.17)

$$L = A\,\mathbf{diag}(w)A^T$$

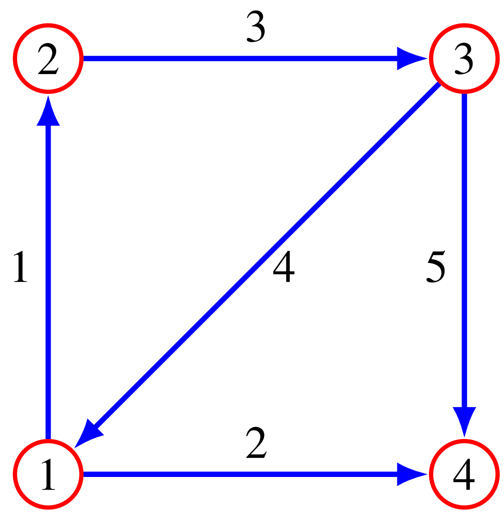
- we number the edges 1 to m
- we make the graph directed by giving each edge an (arbitrary) orientation
- A is the $n \times m$ incidence matrix of the directed graph

$$A_{ik} = \begin{cases} -1 & \text{directed edge } k \text{ points from vertex } i \\ 1 & \text{directed edge } k \text{ points at vertex } i \\ 0 & \text{otherwise} \end{cases}$$

- w is the m-vector of edge weights

$$w_k = W_{ij} \quad \text{if edge } k \text{ points from vertex } j \text{ to vertex } i$$

Example



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

$$\begin{aligned} A\,\mathbf{diag}(w)A^T &= \begin{bmatrix} w_1+w_2+w_4 & -w_1 & -w_4 & -w_2 \\ -w_1 & w_1+w_3 & -w_3 & 0 \\ -w_4 & -w_3 & w_3+w_4+w_5 & -w_5 \\ -w_2 & 0 & -w_5 & w_2+w_5 \end{bmatrix} \\ &= \begin{bmatrix} \deg(1) & -W_{12} & -W_{13} & -W_{14} \\ -W_{21} & \deg(2) & -W_{23} & -W_{24} \\ -W_{31} & -W_{32} & \deg(3) & -W_{34} \\ -W_{41} & -W_{42} & -W_{43} & \deg(4) \end{bmatrix} \end{aligned}$$

Laplacian quadratic form

$$x^T L x = \sum_{\{i,j\} \in E} W_{ij}(x_i - x_j)^2$$

(see derivation on next page)

- x is an n-vector, x_i is a scalar quantity associated with vertex i
- $x^T L x$ is small if entries of x at adjacent vertices are close to each other
- each edge appears once in this sum
- other equivalent expressions are

$$\begin{aligned} x^T L x &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} W_{ij}(x_i - x_j)^2 \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}(x_i - x_j)^2 \end{aligned}$$

the formula for $x^T L x$ can be verified in several ways

- from the definition $L = \mathbf{diag}(W\mathbf{1}) - W$:

$$
\begin{aligned}
x^T L x &= \sum_{i=1}^{n} (\sum_{j=1}^{n} W_{ij}) x_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} x_i x_j \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (x_i^2 - x_i x_j) \\
&= \sum_{i=1}^{n} \sum_{j=i+1}^{n} W_{ij} (x_i^2 - 2 x_i x_j + x_j^2) \\
&= \sum_{i=1}^{n} \sum_{j=i+1}^{n} W_{ij} (x_i - x_j)^2
\end{aligned}
$$

- from the Gram matrix expression $L = A\, \mathbf{diag}(w) A^T$:

$$
x^T L x = \sum_{k=1}^{m} w_k (A^T x)_k^2 = \sum_{k=1}^{m} w_k (x_{i_k} - x_{j_k})^2
$$

  if in the directed graph edge k is oriented from vertex j_k to i_k

Matrix extension

suppose X is an $n \times p$ matrix with rows $x_1^T, \ldots, x_n^T$

$$\operatorname{trace}(X^T L X) = \sum_{\{i,j\} \in E} W_{ij} \|x_i - x_j\|^2$$

- here we associate a vector x_i with vertex i
- $\operatorname{trace}(X^T L X)$ is small if distances of vectors at adjacent vertices are small
- follows from formula for Laplacian quadratic form applied to the columns of X:

$$\operatorname{trace}(X^T L X) = \sum_{k=1}^{p} (Xe_k)^T L (Xe_k) = \sum_{k=1}^{p} \sum_{\{i,j\} \in E} W_{ij} (X_{ik} - X_{jk})^2$$

- other expressions:

$$\begin{aligned} \operatorname{trace}(X^T L X) &= \sum_{i=1}^{n} \sum_{j=i+11}^{n} W_{ij} \|x_i - x_j\|^2 \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|x_i - x_j\|^2 \end{aligned}$$

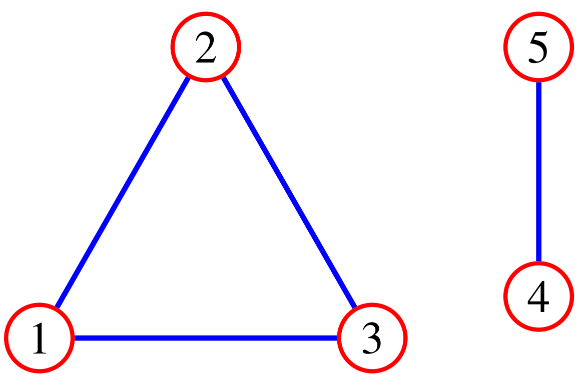
Rank and nullspace

the following properties were shown in lecture 2 and homework 1

- the graph Laplacian L is positive semidefinite
- the rank of L is n minus the number of connected components in the graph
- if the graph is connected, the nullspace of L is spanned by the n-vector $\mathbf{1}$
- if the graph has c connected components, nullspace is $\operatorname{span}(y_1, \ldots, y_c)$, where

$$(y_k)_i = \begin{cases} 1 & \text{vertex } i \text{ is in connected component } k \\ 0 & \text{otherwise} \end{cases}$$

Example



$$\operatorname{rank}(L) = 3, \quad \operatorname{null}(L) = \operatorname{span}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right)$$

Outline

- Laplacian matrix
- **graph partitioning**
- spectral clustering

Vertex partition

Vertex partition

- a vertex partition is a collection of nonempty subsets $V_1, \ldots, V_K$ of V with

$$V = V_1 \cup \cdots \cup V_K, \qquad V_i \cap V_j = \emptyset \quad \text{for } i \neq j$$

- a partition with two subsets V_1 and $V_2 = V \setminus V_1$ is called a *cut*

Value of a cut

$$\mathrm{cut}(V_k) = \sum_{i \in V_k, j \notin V_k} W_{ij}$$

- sum of the weights of the edges connecting vertices in V_k to vertices outside V_k
- with this notation, the total weight of edges between subsets of the partition is

$$\frac{1}{2} \sum_{k=1}^{K} \mathrm{cut}(V_k)$$

Weight of a subgraph

- we give a positive weight d_i to each vertex i

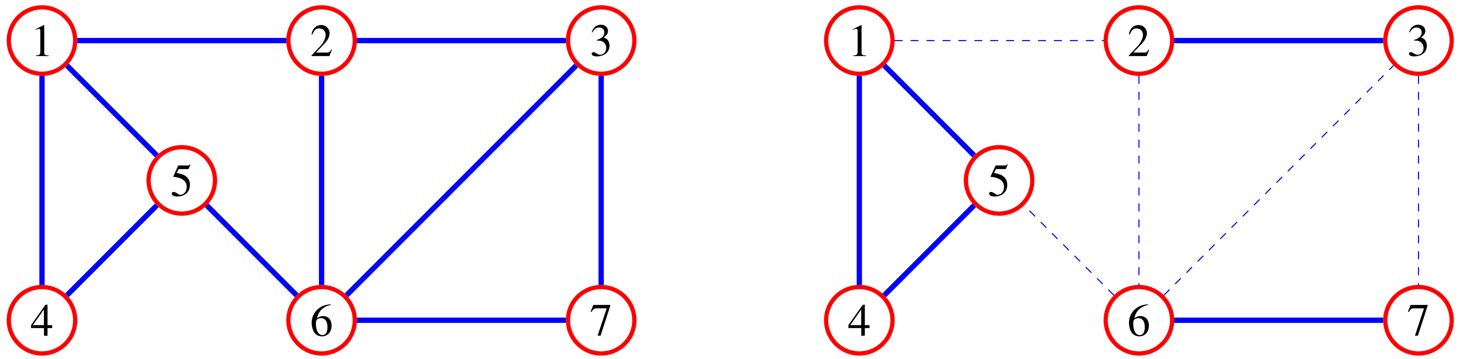
- the total weight of a subset V_k in the partition is denoted by

$$\operatorname{size}(V_k) = \sum_{i \in V_k} d_i$$

- if $d_i = 1$, then $\operatorname{size}(V_k)$ is simply the number of vertices in V_k

- another common choice of vertex weight is the degree: $d_i = \deg(i)$

Example



vertex partition with three sets $V_1 = \{1, 4, 5\}$, $V_2 = \{2, 3\}$, $V_3 = \{6, 7\}$

$$
\begin{aligned}
\mathrm{cut}(V_1) &= W_{12} + W_{56} & \mathrm{size}(V_1) &= d_1 + d_4 + d_5 \\
\mathrm{cut}(V_2) &= W_{12} + W_{26} + W_{36} + W_{37} & \mathrm{size}(V_2) &= d_2 + d_3 \\
\mathrm{cut}(V_3) &= W_{56} + W_{26} + W_{36} + W_{37} & \mathrm{size}(V_3) &= d_6 + d_7
\end{aligned}
$$

Clustering objective

to evaluate the quality of a partition $V_1, \ldots, V_k$ we define the cost function

$$\sum_{k=1}^{K} \frac{\mathrm{cut}(V_k)}{\mathrm{size}(V_k)}$$

- $\mathrm{cut}(V_k)$ is the total weight of edges between V_k and $V \setminus V_k$
- dividing by $\mathrm{size}(V_k)$ discourages using small sets V_k in the partition
- with vertex weights $d_i = 1$, this is called the *ratio cut* objective
- with vertex weights $d_i = \deg(i)$, it is called the *normalized cut* objective

- finding a partition with minimum cost is a hard combinatorial problem
- spectral clustering uses eigendecompositions to find approximate solutions

Outline

- Laplacian matrix
- graph partitioning
- **spectral clustering**

Indicator vector

Indicator vector

- an n-vector with elements 0 and 1
- indicator vector x indicates membership of a subset $S \subseteq V$:

$$x_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

Normalization

- we'll call a positive multiple of an indicator vector a *scaled* indicator vector
- the scaling of a scaled indicator vector x will be defined via a normalization

$$x^T \mathbf{diag}(d) x = \sum_{i=1}^{n} d_i x_i^2 = 1$$

- with this normalization (and using notation $\mathrm{size}(S) = \sum_{i \in S} d_i$),

$$x_i = \begin{cases} 1/\sqrt{\mathrm{size}(S)} & i \in S \\ 0 & i \notin S \end{cases}$$

Indicator matrix

we represent a vertex partition by an $n \times K$ indicator matrix X:

1. columns are scaled indicator vectors (defining K subsets $V_1, \ldots, V_K$ of V)
2. columns are scaled so that nonzero in column k is $1/\sqrt{\mathrm{size}(V_k)}$

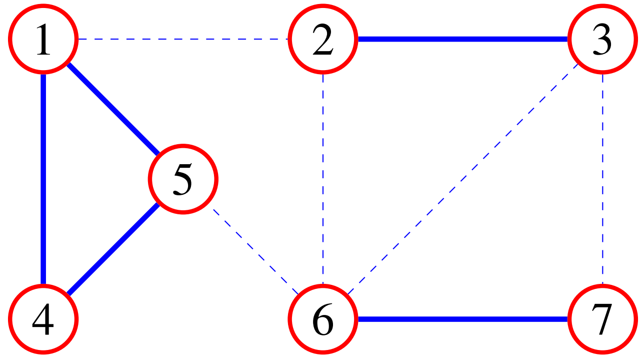
$$X_{ik} = \begin{cases} 1/\sqrt{\mathrm{size}(V_k)} & i \in V_k \\ 0 & \text{otherwise} \end{cases}$$

3. columns are mutually orthogonal ($V_i \cap V_j = \emptyset$ for $i \neq j$)
4. no row is zero ($V_1 \cup \cdots \cup V_K = V$)

if property 1 holds, properties 2 and 3 can be summarized as

$$X^T \mathbf{diag}(d) X = I$$

Example



indicator matrix for this partition, with unit vertex weights $d_i = 1$

$$X = \begin{bmatrix} 1/\sqrt{3} & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 0 & 0 \\ 1/\sqrt{3} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix}$$

Clustering objective

suppose X is an indicator matrix (satisfying the four properties on page 7.16)

- if x_i^T and x_j^T are two rows of X, then

$$\|x_i - x_j\|^2 = \begin{cases} 0 & \text{vertices } i \text{ and } j \text{ are in the same subset} \\ \dfrac{1}{\mathrm{size}(V_k)} + \dfrac{1}{\mathrm{size}(V_l)} & i \in V_k,\ j \in V_l, \text{ and } k \neq l \end{cases}$$

- the clustering objective of page 7.14 can be written as $\mathrm{trace}(X^T L X)$:

$$\begin{aligned} \mathrm{trace}(X^T L X) &= \sum_{\{i,j\} \in E} W_{ij} \|x_i - x_j\|^2 \\ &= \sum_{k=1}^{K} \sum_{i \in V_k, j \notin V_k} \frac{W_{ij}}{\mathrm{size}(V_k)} \\ &= \sum_{k=1}^{K} \frac{\mathrm{cut}(V_k)}{\mathrm{size}(V_k)} \end{aligned}$$

Optimal partition

to summarize, optimal partitions are solutions X of the optimization problem

$$
\begin{array}{ll}
\text{minimize} & \operatorname{trace}(X^T L X) \\
\text{subject to} & X^T \mathbf{diag}(d) X = I \\
& \text{columns of } X \text{ are scaled indicator vectors} \\
& X \text{ has no zero rows}
\end{array}
$$

- the $n \times K$ matrix X is an indicator matrix of the partition
- the second constraint makes this a difficult combinatorial problem
- to simplify the problem we omit the difficult constraints
- the simpler problem is called a *relaxation* of the difficult problem
- we solve the relaxation and round its solution to a suboptimal indicator matrix X

Spectral clustering for ratio cut objective

first consider the relaxed problem with vertex weights $d_i = 1$:

$$\begin{array}{ll} \text{minimize} & \operatorname{trace}(X^T L X) \\ \text{subject to} & X^T X = I \end{array}$$

- solution follows from eigendecomposition of Laplacian

$$L = Q \Lambda Q^T = \sum_{i=1}^{n} \lambda_i q_i q_i^T$$

- columns of optimal $\tilde{X}$ are last K eigenvectors (for smallest K eigenvalues):

$$X = \left[\begin{array}{ccc} q_{n-K+1} & \cdots & q_n \end{array} \right]$$

- if the graph is connected, $\mathbf{1}$ is in the range of X, so X has no zero rows

optimal solution of relaxed problem is not necessarily a valid indicator matrix

k-means rounding

to find a valid partition $V_1, \ldots, V_K$ from the solution X of the relaxed problem:

- apply the k-means algorithm (with $k = K$) to the n rows of X
- the result is a clustering of the rows in K groups with representatives $s_1, \ldots, s_K$
- assign vertex i to set V_k if row i of X is assigned to the cluster of s_k

Motivation for k-means rounding

the k-means rounding method may be justified as follows

- k-means applied to the rows of X computes an approximate factorization

$$X \approx \tilde{X}\tilde{S}$$

- $\tilde{X}$ is an $n \times K$ indicator matrix (elements in column k are 0 and $1/\sqrt{\mathrm{size}(V_k)}$)
- $\tilde{S}$ is a $K \times K$ matrix; rows are scaled representatives $\sqrt{\mathrm{size}(V_k)}s_k^T$
- since $X^T X = I$, the matrix $\tilde{S}$ is approximately orthogonal:

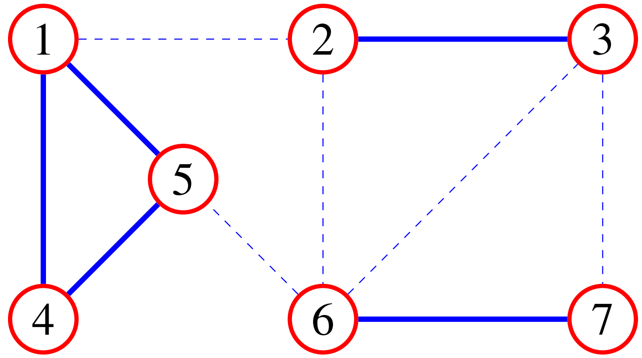
$$I = X^T X \approx \tilde{S}^T \tilde{X}^T \tilde{X}\tilde{S} = \tilde{S}^T \tilde{S}$$

- therefore $\tilde{X} \approx X\tilde{S}^T$ is an indicator matrix with clustering objective

$$\mathrm{trace}(\tilde{X}^T L\tilde{X}) \approx \mathrm{trace}(\tilde{S}X^T LX\tilde{S}^T) \approx \mathrm{trace}(X^T LX)$$

i.e., close to the optimal value of the relaxed optimization problem

Example



suppose k-means applied to the rows of the solution X of the relaxation gives

$$X \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} = \tilde{X}\tilde{S}, \quad \tilde{X} = \begin{bmatrix} 1/\sqrt{3} & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 0 & 0 \\ 1/\sqrt{3} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix}, \quad \tilde{S} = \begin{bmatrix} \sqrt{3}s_1^T \\ \sqrt{2}s_2^T \\ \sqrt{2}s_3^T \end{bmatrix}$$

we take partition indicated by $\tilde{X}$ as approximate solution of partitioning problem

Spectral clustering for normalized cut

the relaxed problem with vertex weights $d_i = \deg(i)$ is

$$\begin{array}{ll} \text{minimize} & \mathrm{trace}(X^T L X) \\ \text{subject to} & X^T \,\mathbf{diag}(d) X = I \end{array}$$

- solution follows from generalized eigendecomposition of L, $\mathbf{diag}(d)$
- solution is $X = \mathbf{diag}(d)^{-1/2} Y$ where Y is the solution of

$$\begin{array}{ll} \text{minimize} & \mathrm{trace}(Y^T L_\mathrm{n} Y) \\ \text{subject to} & Y^T Y = I \end{array}$$

  and L_n is the normalized Laplacian (page 7.4)

$$L_\mathrm{n} = \mathbf{diag}(d)^{-1/2} L \,\mathbf{diag}(d)^{-1/2}$$

- columns of optimal Y are the last K eigenvectors of L_n
- we can use k-means to round solution X of relaxation to valid indicator matrix

Example

- participants in a study are asked to score 24 animals on a list of 764 properties[1]
- the result is a 764×24 table of scores from 0 to 4

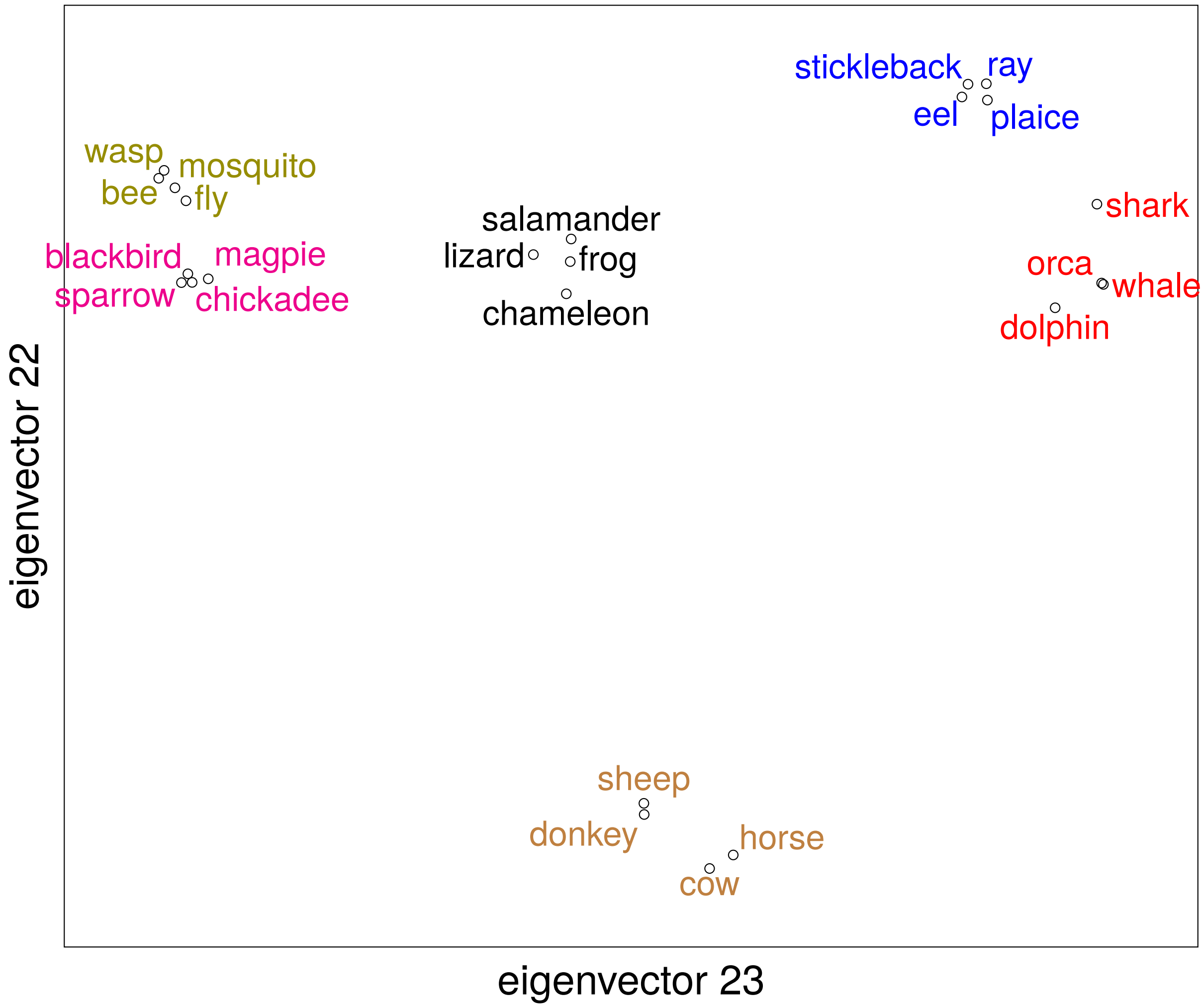
	bee	donkey	shark	frog	sparrow	$\cdots$
is dangerous	2	0	4	0	0	$\cdots$
has a tail	0	4	2	1	2	$\cdots$
lives in the woods	3	0	0	2	3	$\cdots$
is beautiful	0	2	1	0	2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

- cosine similarities of columns give a semantic similarity between the 24 names
- we define a graph with 24 vertices and the cosine similarities as edge weights

[1]Liuzzi, A. G. *et al.*, *Cross-modal representation of spoken and written word meaning in left pars triangularis*, NeuroImage (2017).

Spectral clustering with normalized ratio cut

- the figure shows the entries of the generalized eigenvectors 22 and 23 of L
- the six clusters are found by k-means with $K = 6$



References

- Ulrike von Luxburg, *A tutorial on spectral clustering*, Statistics and Computing (2007).

  the methods we discussed are algorithms 1 and 2 on page 399

- Jianbo Shi and Jitendra Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000).

  discusses the generalized eigenvalue method for normalized cut objective