Drawing Large Graphs Using Divisive Hierarchical k-means

Barbara Ikica

22. 11. 2012

Abstract

- Idea behind the algorithm

Abstract

- Idea behind the algorithm
- Implementation

Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)

Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation

Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation
  - Identifying connected components

Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation
  - Identifying connected components
  - Diffusion kernels on graphs

Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation
  - Identifying connected components
  - Diffusion kernels on graphs
  - Drawing

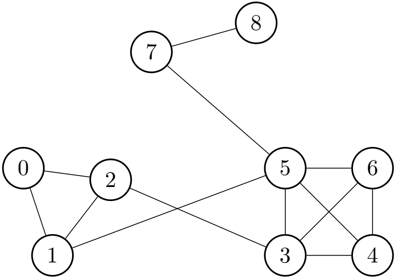
Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation
  - Identifying connected components
  - Diffusion kernels on graphs
  - Drawing
  - Time complexity

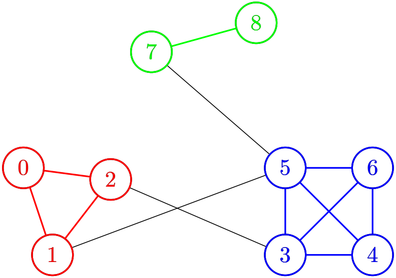
Abstract

- Idea behind the algorithm
- Implementation
  - Hierarchical clustering (Divisive Hierarchical k-means)
  - Graph representation
  - Identifying connected components
  - Diffusion kernels on graphs
  - Drawing
  - Time complexity
  - Random projections

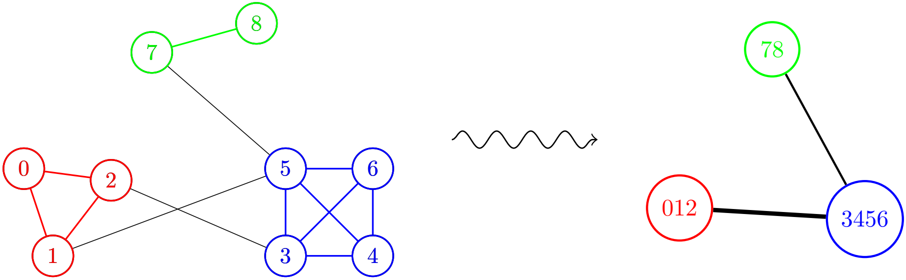
Idea behind the algorithm



Idea behind the algorithm



Idea behind the algorithm



Idea behind the algorithm

Objective: Arrangement of the (sets of the) vertices of a graph $G = \big(V(G), E(G)\big)$ at the level of hierarchy n

Idea behind the algorithm

Objective: Arrangement of the (sets of the) vertices of a graph $G = \big(V(G), E(G)\big)$ at the level of hierarchy n

1. Determining the partition of $V(G)$: $\mathcal{P}_n = \{M_i\}_{i=1}^m$

Idea behind the algorithm

Objective: Arrangement of the (sets of the) vertices of a graph $G = \big(V(G), E(G)\big)$ at the level of hierarchy n

1. Determining the partition of $V(G)$: $\mathcal{P}_n = \{M_i\}_{i=1}^m$
2. Determining the drawing area for each $M_i \in \mathcal{P}_n$

Implementation – Hierarchical clustering

Implementation – Hierarchical clustering

k-means clustering

$M := \{v_i\}_{i=1}^{n}, \quad v_i \in \mathbb{R}^d \quad \forall i, \quad 1 \leq i \leq n.$
We aim to partition the set M into k *clusters* $\mathcal{P} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares

$$\boxed{\min_{\mathcal{P}} \sum_{i=1}^{k} \sum_{v_j \in S_i} \left\| v_j - \mu_i \right\|^2},$$

where $\mu_i := \frac{1}{|S_i|}\left(\sum_{v_j \in S_i} v_j\right)$ is the *centroid vector* of the cluster S_i.

Implementation – Hierarchical clustering

k-means clustering

- NP-hard problem

Implementation – Hierarchical clustering

k-means clustering

- NP-hard problem
- If k and d are fixed, the problem can be exactly solved in $O(n^{dk+1} \log n)$ steps.

Implementation – Hierarchical clustering

k-means clustering

Recall that:

We aim to partition the set M into k *clusters* $\mathcal{P} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares

$$\boxed{\min_{\mathcal{P}} \sum_{i=1}^{k} \sum_{v_j \in S_i} \left\| v_j - \mu_i \right\|^2}.$$

Corollary 1

$$\min_{z \in \mathbb{R}^d} \sum_{v \in S} \|v - z\|^2 = \sum_{v \in S} \|v - \mu\|^2$$

Implementation – Hierarchical clustering

k-means clustering

Lemma 1

Choose arbitrary $S \subset \mathbb{R}^d$ and $z \in \mathbb{R}^d$. Then

$$\sum_{v \in S} \|v - z\|^2 = \sum_{v \in S} \|v - \mu\|^2 + |S| \|z - \mu\|^2,$$

where μ is the centroid vector of the cluster S, i.e. $\mu = \frac{1}{|S|} \sum_{v \in S} v$.

Corollary 1

$$\min_{z \in \mathbb{R}^d} \sum_{v \in S} \|v - z\|^2 = \sum_{v \in S} \|v - \mu\|^2$$

Implementation – Hierarchical clustering

k-means clustering

Lemma 1

Choose arbitrary $S \subset \mathbb{R}^d$ and $z \in \mathbb{R}^d$. Then

$$\sum_{v \in S} \|v - z\|^2 = \sum_{v \in S} \|v - \mu\|^2 + |S|\|z - \mu\|^2,$$

where μ is the centroid vector of the cluster S, i.e. $\mu = \frac{1}{|S|} \sum_{v \in S} v$.

Lemma 2

Let X denote an arbitrary random variable with values in $\mathbb{R}^d$. For any $z \in \mathbb{R}^d$ the following holds:

$$E(\|X - z\|^2) = E(\|X - E(X)\|^2) + \|z - E(X)\|^2.$$

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

- Randomly pick k vectors from $M = \{v_1, v_2, \ldots, v_n\}$ as the centroids $\mu_i^{(0)}$ for each $i = 1, 2, \ldots, k$.

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

- Randomly pick k vectors from $M = \{v_1, v_2, \ldots, v_n\}$ as the centroids $\mu_i^{(0)}$ for each $i = 1, 2, \ldots, k$.
- For $t = 0, 1, 2, \ldots$ repeat (for each $i = 1, 2, \ldots, k$)
  1. $S_i^{(t)} = \left\{ v_p : \left\| v_p - \mu_i^{(t)} \right\|^2 \leq \left\| v_p - \mu_j^{(t)} \right\|^2 \quad \forall\, j : 1 \leq j \leq k \right\}$

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

- Randomly pick k vectors from $M = \{v_1, v_2, \ldots, v_n\}$ as the centroids $\mu_i^{(0)}$ for each $i = 1, 2, \ldots, k$.
- For $t = 0, 1, 2, \ldots$ repeat (for each $i = 1, 2, \ldots, k$)
  1. $S_i^{(t)} = \left\{ v_p : \left\| v_p - \mu_i^{(t)} \right\|^2 \leq \left\| v_p - \mu_j^{(t)} \right\|^2 \quad \forall\, j : 1 \leq j \leq k \right\}$ and
  2. $\mu_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{v_j \in S_i^{(t)}} v_j$

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

- Randomly pick k vectors from $M = \{v_1, v_2, \ldots, v_n\}$ as the centroids $\mu_i^{(0)}$ for each $i = 1, 2, \ldots, k$.
- For $t = 0, 1, 2, \ldots$ repeat (for each $i = 1, 2, \ldots, k$)
  1. $S_i^{(t)} = \left\{ v_p : \left\| v_p - \mu_i^{(t)} \right\|^2 \leq \left\| v_p - \mu_j^{(t)} \right\|^2 \quad \forall\, j : 1 \leq j \leq k \right\}$ and
  2. $\mu_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{v_j \in S_i^{(t)}} v_j$

  until there is no further change in the assignments of the vectors to the clusters $S_i^{(t)}$ (for each i) in the partition V.

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k}\sum_{v_j\in S_i^{(t)}}\left\|v_j-\mu_i^{(t)}\right\|^2$ *decreases monotonically during iteration.*

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof.

□

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof.

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t-1)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \tag{1}$$

□

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof.

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t-1)}} \left\| v_j - \mu_i^{(t)} \right\|^2, \text{ since} \qquad (1)$$

$$S_i^{(t)} = \left\{ v_p : \left\| v_p - \mu_i^{(t)} \right\|^2 \leq \left\| v_p - \mu_j^{(t)} \right\|^2 \quad \forall\, j : 1 \leq j \leq k \right\}.$$

□

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof.

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \tag{2}$$

□

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof.

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2, \text{ since} \quad (2)$$

$$\mu_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{v_j \in S_i^{(t)}} v_j \quad \text{and} \quad \min_{z \in \mathbb{R}^d} \sum_{v \in S} \|v - z\|^2 = \sum_{v \in S} \|v - \mu\|^2.$$

□

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof. Thus

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t-1)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \tag{1}$$

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \tag{2}$$

□

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof. Thus

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t+1)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \tag{1}$$

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2 \tag{2}$$

□

Implementation – Hierarchical clustering

k-means clustering – iterative algorithm

Lemma 3

The value of the expression $\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2$ *decreases monotonically during iteration.*

Proof. Thus

$$\sum_{i=1}^{k} \sum_{v_j \in S_i^{(t+1)}} \left\| v_j - \mu_i^{(t+1)} \right\|^2 \leq \sum_{i=1}^{k} \sum_{v_j \in S_i^{(t)}} \left\| v_j - \mu_i^{(t)} \right\|^2.$$

□

Implementation – Hierarchical clustering

Hierarchical clustering

Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

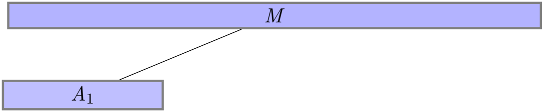


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

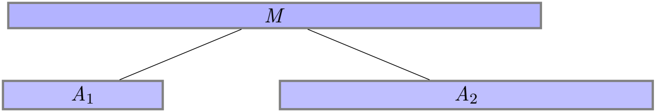


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

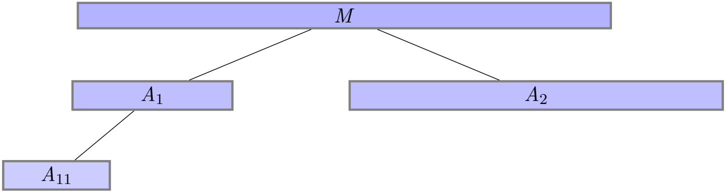


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

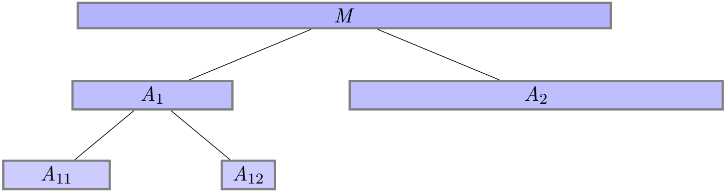


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

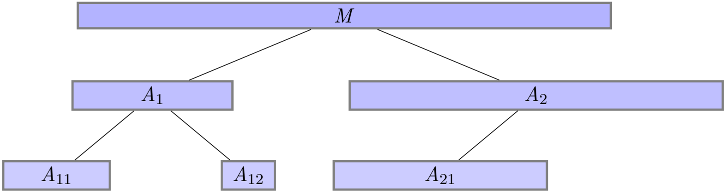


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

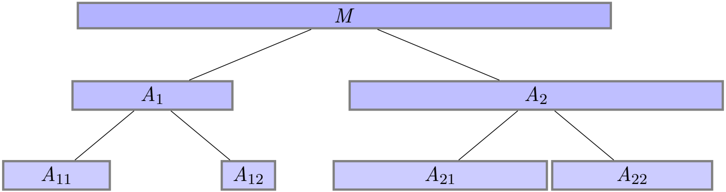


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

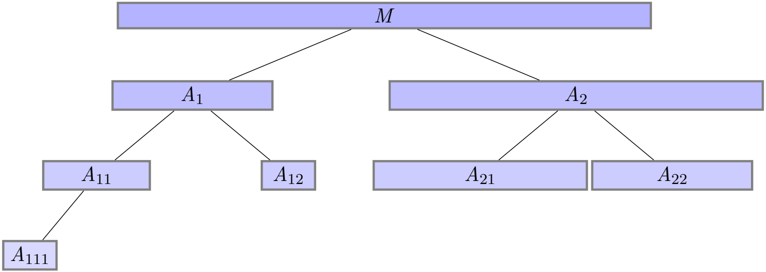


Figure: Schematic representation of the hierarchical partition of the set M.

Hierarchical clustering

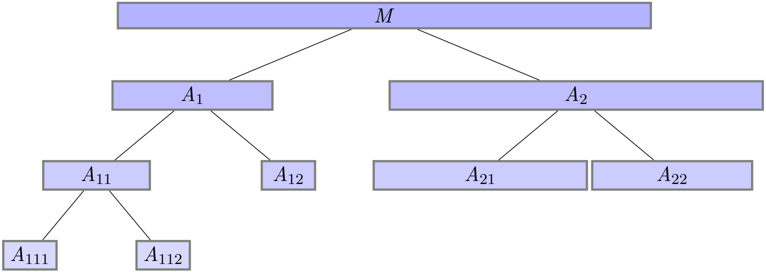


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

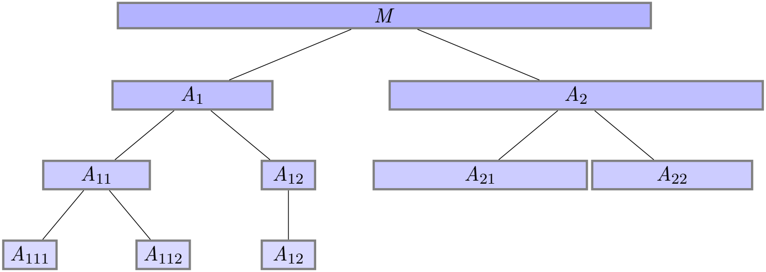


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

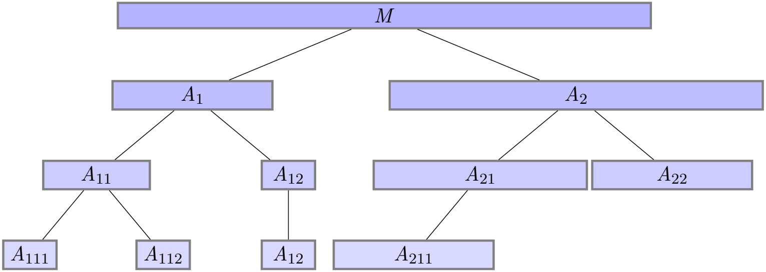


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

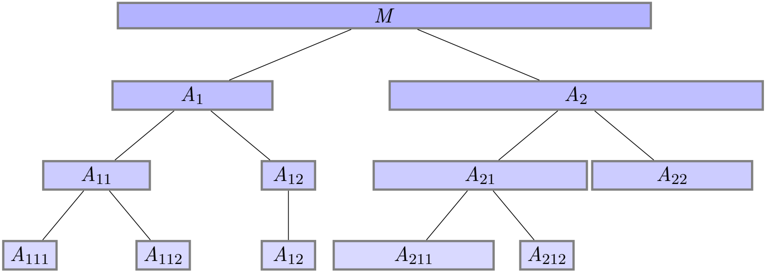


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Hierarchical clustering

Hierarchical clustering

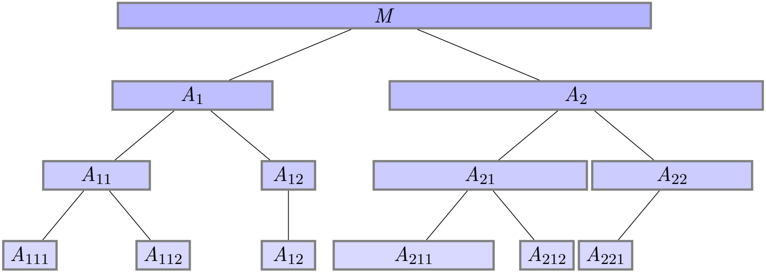


Figure: Schematic representation of the hierarchical partition of the set M.

Hierarchical clustering

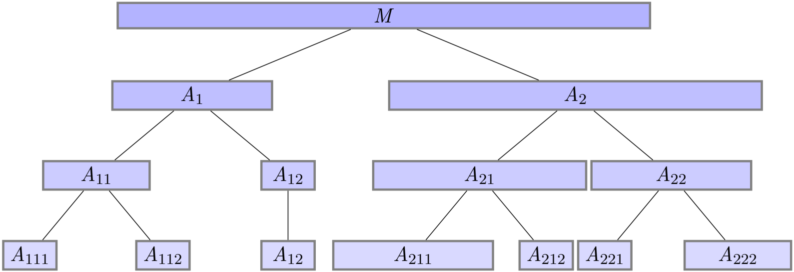


Figure: Schematic representation of the hierarchical partition of the set M.

Implementation – Graph representation

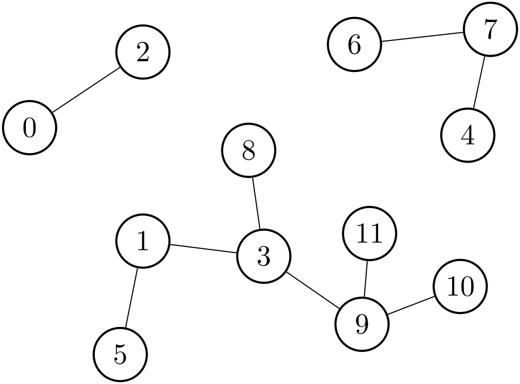
Input (graph data)

```
0 2
1 3
1 5
3 8
3 9
4 7
6 7
9 10
9 11
```

Implementation – Graph representation

Input (graph data)

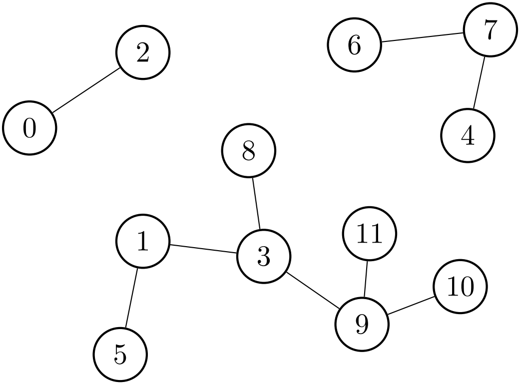
```
0  2
1  3
1  5
3  8
3  9
4  7
6  7
9  10
9  11
```



Implementation – Graph representation

Input (graph data)

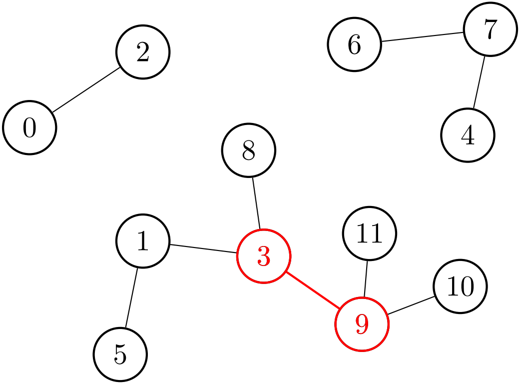
```
0  2
1  3
1  5
3  8
3  9
4  7
6  7
9  10
9  11
```



Implementation – Graph representation

Input (graph data)

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



Implementation – Graph representation

Adjacency matrix M^G

$$[M^G]_{ij} = \begin{cases} 1; & v_i \sim v_j, \\ 0; & otherwise. \end{cases}$$

Implementation – Graph representation

Adjacency matrix M^G

$$[M^G]_{ij} = \begin{cases} 1; & v_i \sim v_j, \\ 0; & otherwise. \end{cases}$$

Example

$$M^G \quad = \quad \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

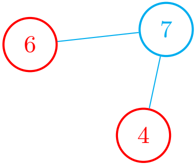
Implementation – Graph representation

Adjacency matrix M^G

$$[M^G]_{ij} = \begin{cases} 1; & v_i \sim v_j, \\ 0; & otherwise. \end{cases}$$

Example

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$



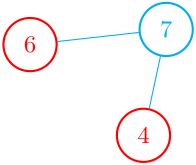
Implementation – Graph representation

Adjacency matrix M^G

$$[M^G]_{ij} = \begin{cases} 1; & v_i \sim v_j, \\ 0; & otherwise. \end{cases}$$

Example

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$



Implementation – Graph representation

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

Implementation – Graph representation

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix} \longrightarrow \begin{bmatrix} [2] \\ [3,5] \\ [0] \\ [1,8,9] \\ [7] \\ [1] \\ [7] \\ [4,6] \\ [3] \\ [3,10,11] \\ [9] \\ [9] \end{bmatrix}$$

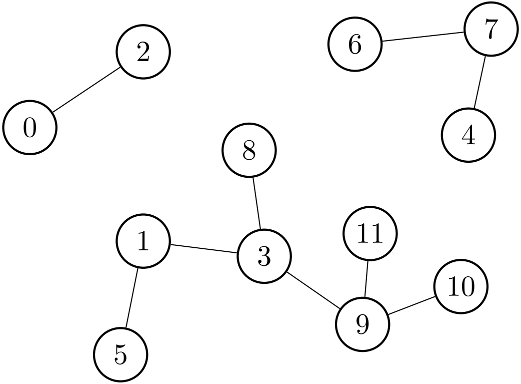
Implementation – Graph representation

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix} \longrightarrow \begin{bmatrix} [2] \\ [3,5] \\ [0] \\ [1,8,9] \\ [7] \\ [1] \\ [7] \\ [4,6] \\ [3] \\ [3,10,11] \\ [9] \\ [9] \end{bmatrix}$$

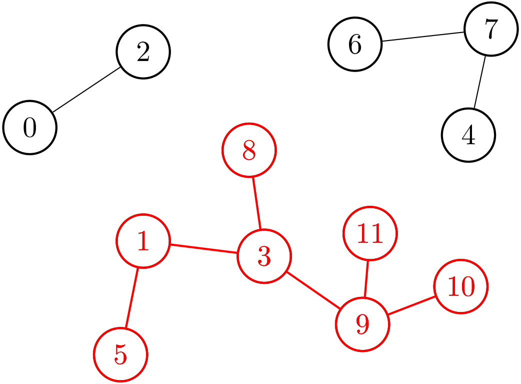
Implementation – Graph representation

$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix} \longrightarrow \begin{bmatrix} [2] \\ [3,5] \\ [0] \\ [1,8,9] \\ [7] \\ [1] \\ [7] \\ [4,6] \\ [3] \\ [3,10,11] \\ [9] \\ [9] \end{bmatrix}$$

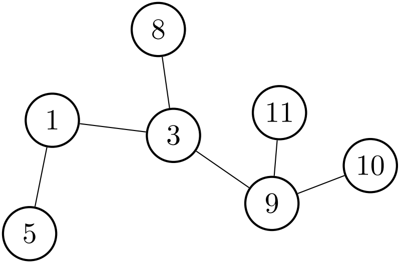
Implementation – Identifying connected components



Implementation – Identifying connected components



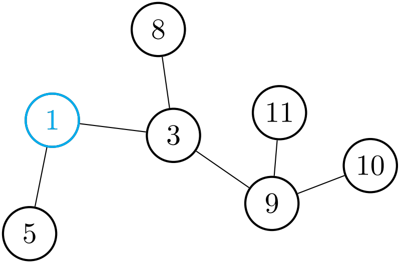
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

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

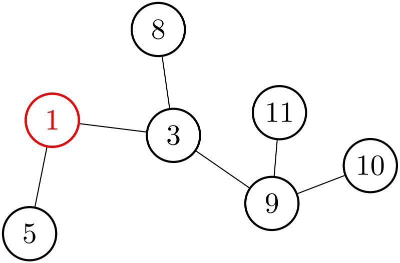
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

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

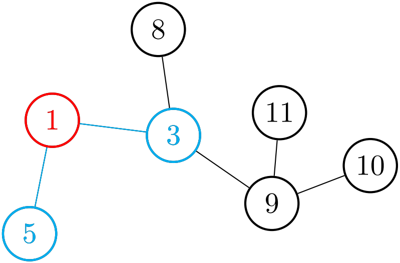
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

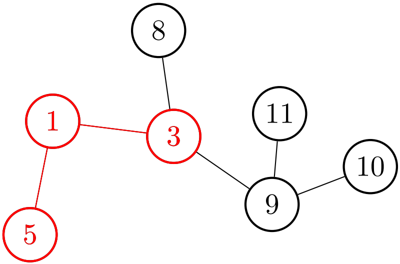
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

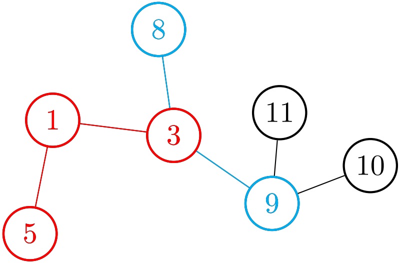
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

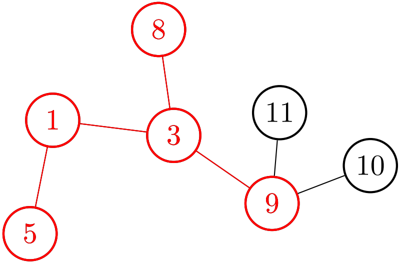
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

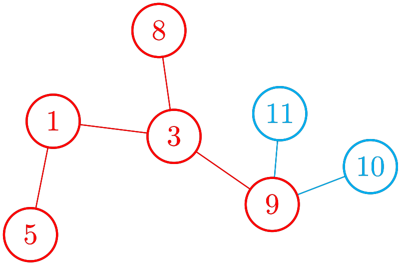
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \end{bmatrix}$$

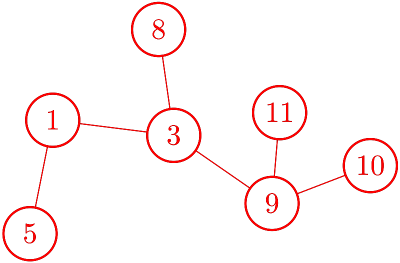
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \end{bmatrix}$$

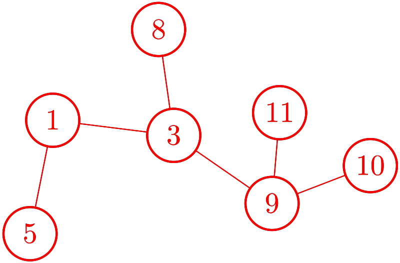
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix}$$

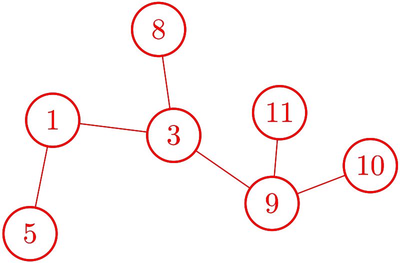
Implementation – Identifying connected components



$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix}$$

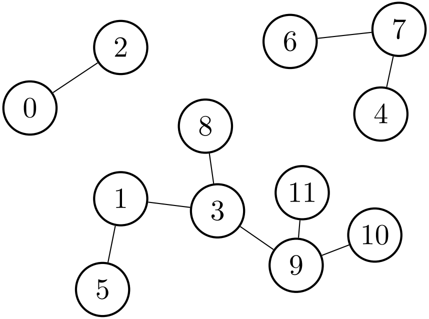
Implementation – Identifying connected components



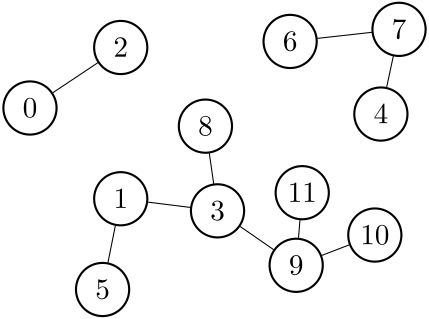
$$M^G = \begin{bmatrix} 0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \end{bmatrix}$$

Implementation – Identifying connected components

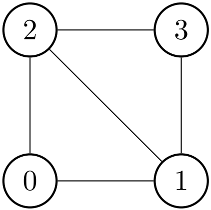


Implementation – Identifying connected components

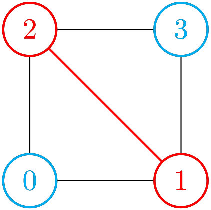


$$\begin{bmatrix}1 & 2 & 1 & 2 & 3 & 2 & 3 & 3 & 2 & 2 & 2 & 2\end{bmatrix}$$

Implementation – Diffusion kernels on graphs



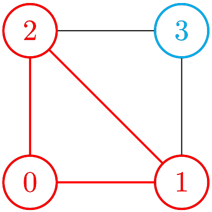
Implementation – Diffusion kernels on graphs



Hierarchical clustering on the row vectors of the adjacency matrix M^G:

$$S_1 = \{1, 2\}$$
$$S_2 = \{0, 3\}$$

Implementation – Diffusion kernels on graphs



Hierarchical clustering on the row vectors of **?**:

$$S_1 = \{0, 1, 2\}$$
$$S_2 = \{3\}$$

Implementation – Diffusion kernels on graphs

The adjacency matrix M^G has the following property:

$[(M^G)^k]_{ij} = \#$ of all paths of length up to k between vertices i and j

Implementation – Diffusion kernels on graphs

The adjacency matrix M^G has the following property:

$[(M^G)^k]_{ij} = \#$ of all paths of length up to k between vertices i and j

We modify the algorithm by replacing M^G with the **kernel matrix** K^G:

$$\boxed{K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)}.$$

Implementation – Diffusion kernels on graphs

- $k : \Omega \times \Omega \to \mathbb{R}$ is a similarity measure if:
  $k(x, y)$ characterizes the similarities of $x, y \in \Omega$.

Implementation – Diffusion kernels on graphs

- $k : \Omega \times \Omega \to \mathbb{R}$ is a similarity measure if:
  $k(x, y)$ characterizes the similarities of $x, y \in \Omega$.
- $k : \Omega \times \Omega \to \mathbb{R}$ is a kernel if:
  1. $k(x, y) = k(y, x)$, $\forall x, y \in \Omega$,
  2. k is *positive semidefinite*:
     the kernel matrix $K \in \mathbb{R}^{n \times n}$, $[K]_{ij} = k(x_i, x_j)$, is positive semidefinite for all $x_1, x_2, \ldots, x_n \in \Omega$.

Implementation – Diffusion kernels on graphs

- $k : \Omega \times \Omega \to \mathbb{R}$ is a similarity measure if:
  $k(x, y)$ characterizes the similarities of $x, y \in \Omega$.
- $k : \Omega \times \Omega \to \mathbb{R}$ is a kernel if:
  1. $k(x, y) = k(y, x)$, $\forall x, y \in \Omega$,
  2. k is *positive semidefinite*:
     the kernel matrix $K \in \mathbb{R}^{n \times n}$, $[K]_{ij} = k(x_i, x_j)$, is positive semidefinite for all $x_1, x_2, \ldots, x_n \in \Omega$.
- Given a kernel k, there exist a Hilbert space $\mathcal{H}_k$ and a map $\phi : \Omega \to \mathcal{H}_k$ such that

$$\langle \phi(x), \phi(y) \rangle_{\mathcal{H}_k} = k(x, y) \text{ for all } x, y \in \Omega.$$

Implementation – Diffusion kernels on graphs

$$K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)$$ indeed is a kernel matrix,

since

Implementation – Diffusion kernels on graphs

$$\boxed{K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)}$$ indeed is a kernel matrix, since

$$K^G = \exp(\alpha M^G)$$

Implementation – Diffusion kernels on graphs

$$\boxed{K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)}$$ indeed is a kernel matrix, since

$$K^G = \exp(\alpha M^G) = \exp(\alpha U D U^T)$$

Implementation – Diffusion kernels on graphs

$\boxed{K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)}$ indeed is a kernel matrix, since

$$K^G = \exp(\alpha M^G) = \exp(\alpha UDU^T) = \alpha U \exp(D) U^T$$

Implementation – Diffusion kernels on graphs

$\boxed{K^G := \sum_{k=0}^{\infty} \frac{\alpha^k (M^G)^k}{k!} = \exp(\alpha M^G)}$ indeed is a kernel matrix, since

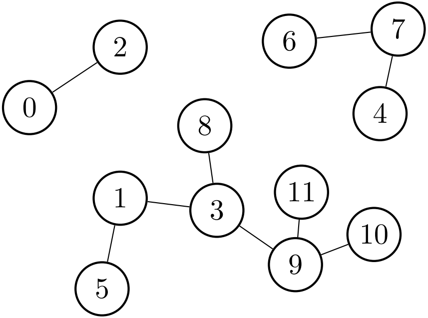
$$K^G = \exp(\alpha M^G) = \exp(\alpha UDU^T) = \alpha U \exp(D) U^T$$

$\implies K^G$ is a symmetric positive semidefinite matrix

Implementation – Hierarchical clustering on K^G

Example

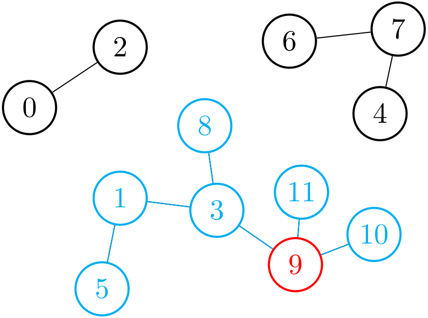
"1#1"
"2#11"
"1#2"
"2#211"
"3#11"
"2#12"
"3#2"
"3#12"
"2#221"
"2#212"
"2#2221"
"2#2222"



Implementation – Hierarchical clustering on K^G

Example

"1#1"
"2#11"
"1#2"
"2#211"
"3#11"
"2#12"
"3#2"
"3#12"
"2#221"
"2#212"
"2#2221"
"2#2222"



Implementation – Hierarchical clustering on K^G

Example

"1#1"

"2#11"

"1#2"

"2#211"

"3#11"

"2#12"

"3#2"

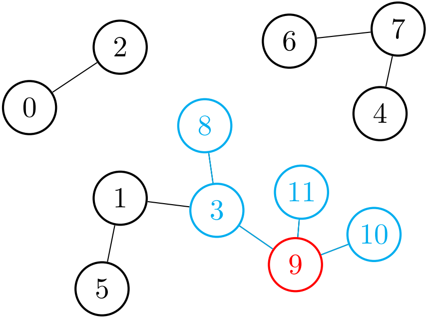
"3#12"

"2#221"

"2#212"

"2#2221"

"2#2222"



Implementation – Hierarchical clustering on K^G

Example

"1#1"

"2#11"

"1#2"

"2#211"

"3#11"

"2#12"

"3#2"

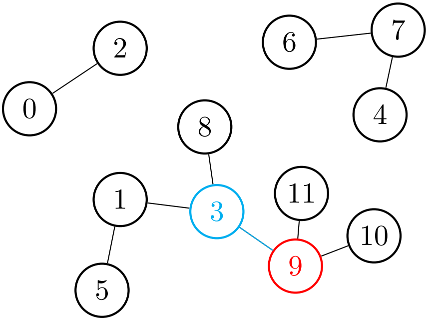
"3#12"

"2#221"

"2#212"

"2#2221"

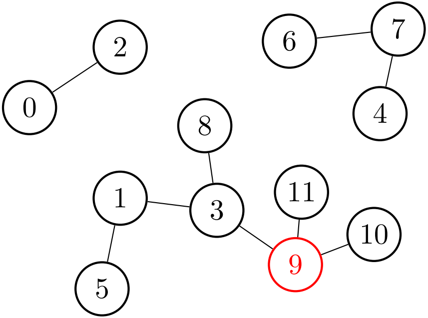
"2#2222"



Implementation – Hierarchical clustering on K^G

Example

"1#1"
"2#11"
"1#2"
"2#211"
"3#11"
"2#12"
"3#2"
"3#12"
"2#221"
"2#212"
"2#2221"
"2#2222"



Implementation – Drawing

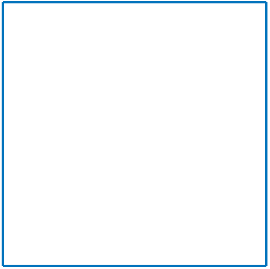


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_0$.

Implementation – Drawing

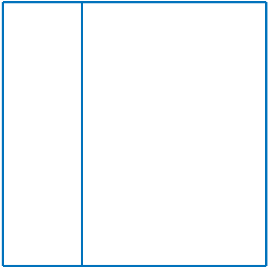
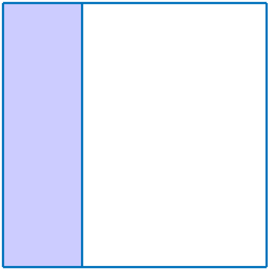


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_1$.

Implementation – Drawing



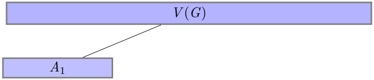


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_1$.

Implementation – Drawing

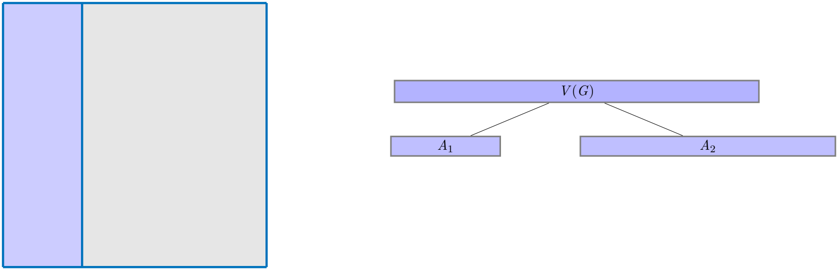


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_1$.

Implementation – Drawing

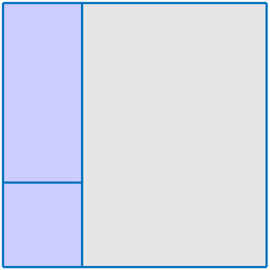
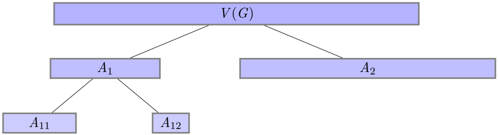


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_2$.



Implementation – Drawing

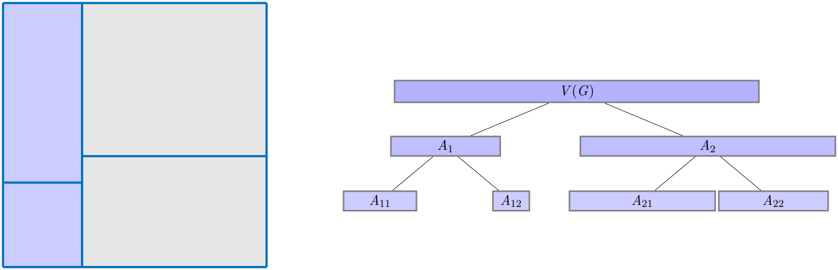


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_2$.

Implementation – Drawing

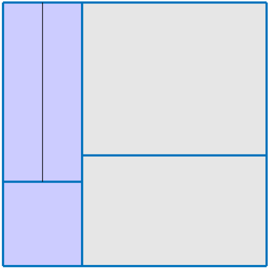
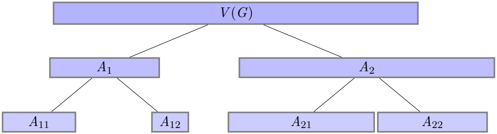


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.



Implementation – Drawing

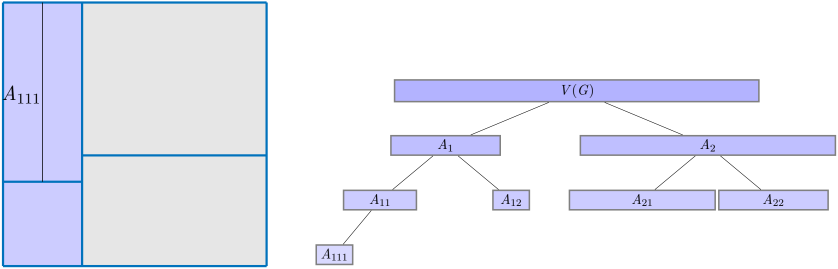
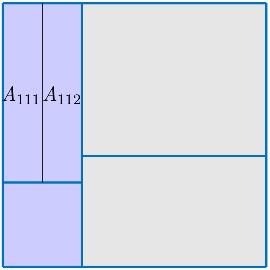


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing



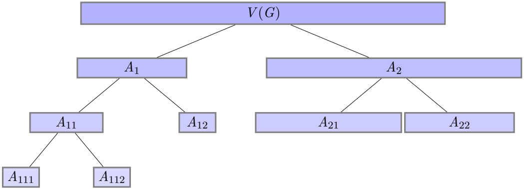
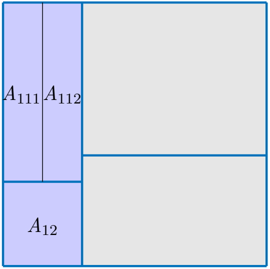


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing



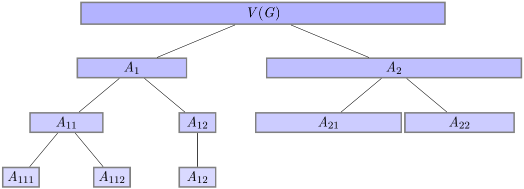


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing

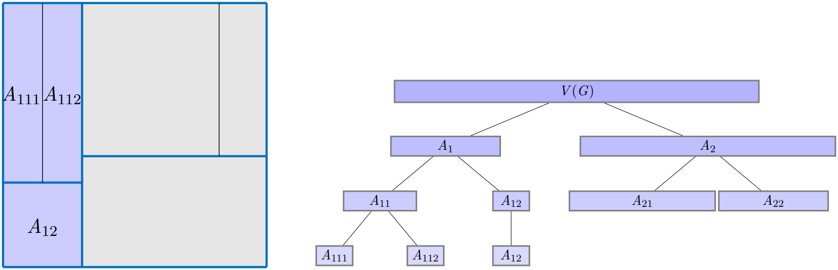


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing

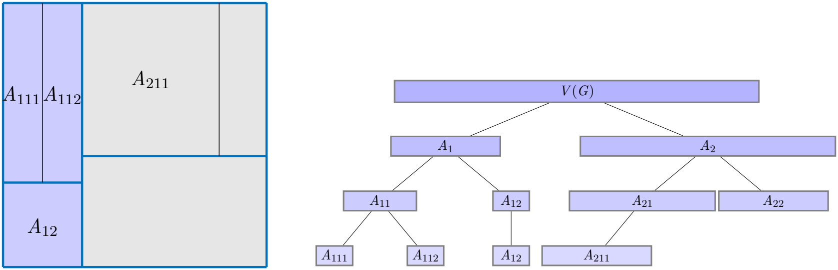


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing

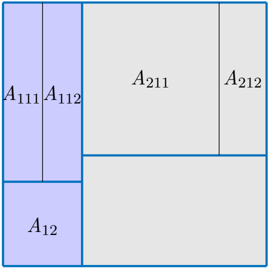
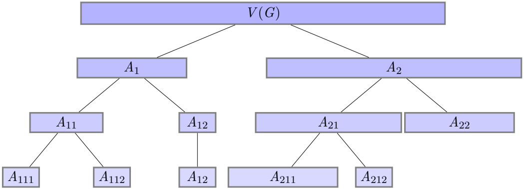
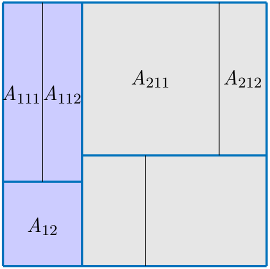


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.



Implementation – Drawing



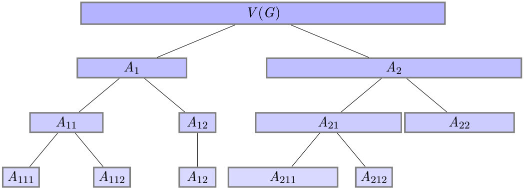


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Drawing

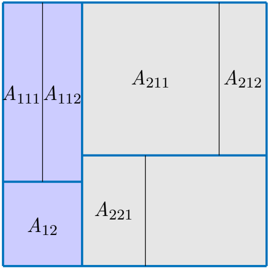
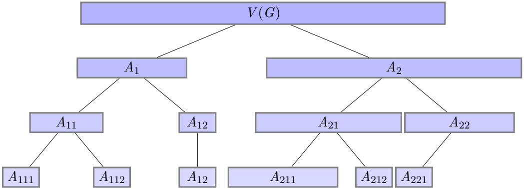


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.



Implementation – Drawing

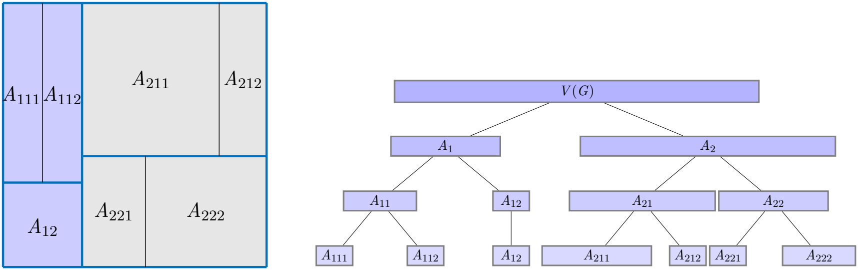


Figure: Determining the drawing area for the sets in the partition $\mathcal{P}_3$.

Implementation – Time complexity

Efficient computation of $K^G = \exp(\alpha M^G)$

Computation of K^G si needed in the algorithm:

- to determine the centroids μ_j,
- to minimize $\|v_i - \mu_j\|^2$

Implementation – Time complexity

Efficient computation of $K^G = \exp(\alpha M^G)$

Computation of K^G si needed in the algorithm:

- to determine the centroids μ_j,
- to minimize $\|v_i - \mu_j\|^2$:

$$\|v_i - \mu_j\|^2 = \langle v_i - \mu_j, v_i - \mu_j \rangle = \langle v_i, v_i \rangle - 2\langle v_i, \mu_j \rangle + \langle \mu_j, \mu_j \rangle.$$

Implementation – Time complexity

Efficient computation of $K^G = \exp(\alpha M^G)$

Computation of K^G si needed in the algorithm:

- to determine the centroids μ_j,
- to minimize $\|v_i - \mu_j\|^2$:

$$\|v_i - \mu_j\|^2 = \langle v_i - \mu_j, v_i - \mu_j \rangle = \langle v_i, v_i \rangle - 2\langle v_i, \mu_j \rangle + \langle \mu_j, \mu_j \rangle.$$

The question is thus:

- How to efficiently multiply the matrix K^G with an arbitrary vector and how to efficiently compute the inner products $\langle v_i, v_i \rangle = \|v_i\|^2$?

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$K^G v = Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!} (M^G)^2 v + \frac{\alpha^3}{3!} (M^G)^3 v + \dots$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$\begin{aligned} K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!} (M^G)^2 v + \frac{\alpha^3}{3!} (M^G)^3 v + \ldots = \\ &= v + \frac{\alpha}{1} (M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1} (M^G v) \right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1} (M^G v) \right) \right) + \ldots \end{aligned}$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$
\begin{aligned}
K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!}(M^G)^2 v + \frac{\alpha^3}{3!}(M^G)^3 v + \ldots = \\
&= v + \frac{\alpha}{1}(M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right)\right) + \ldots
\end{aligned}
$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$
\begin{aligned}
K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!} (M^G)^2 v + \frac{\alpha^3}{3!} (M^G)^3 v + \ldots = \\
&= v + \frac{\alpha}{1}(M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1} (M^G v) \right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1} (M^G v) \right) \right) + \ldots
\end{aligned}
$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$\begin{aligned} K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!}(M^G)^2 v + \frac{\alpha^3}{3!}(M^G)^3 v + \ldots = \\ &= v + \frac{\alpha}{1}(M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right)\right) + \ldots \end{aligned}$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$\begin{aligned} K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!} (M^G)^2 v + \frac{\alpha^3}{3!} (M^G)^3 v + \ldots = \\ &= v + \frac{\alpha}{1}(M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v) \right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v) \right) \right) + \ldots \end{aligned}$$

Implementation – Time complexity

Multiplying the matrix K^G with an arbitrary vector v

$$
\begin{aligned}
K^G v &= Iv + \frac{\alpha}{1} M^G v + \frac{\alpha^2}{2!} (M^G)^2 v + \frac{\alpha^3}{3!} (M^G)^3 v + \ldots = \\
&= v + \frac{\alpha}{1}(M^G v) + \frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right) + \frac{\alpha}{3} M^G \left(\frac{\alpha}{2} M^G \left(\frac{\alpha}{1}(M^G v)\right)\right) + \ldots
\end{aligned}
$$

Implementation – Random projections

Random projections

Corollary of the Johnson–Lindenstrauss lemma:

Theorem 1

Let P be an arbitrary set of n points in $\mathbb{R}^d$, represented as an $n \times d$ matrix $A \in \mathbb{R}^{n \times d}$. Given $\varepsilon > 0, \beta > 0$ let $k_0 = \frac{4+2\beta}{\varepsilon^2/2-\varepsilon^3/3} \log n$. For integer $k \geq k_0$, let $R \in \mathbb{R}^{d \times k}$ be a random matrix with elements r_{ij}, where r_{ij} are independent random variables from either one of the following two probability distributions:

$$r_{ij} : \begin{pmatrix} 1 & -1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad r_{ij} : \begin{pmatrix} 1 & 0 & -1 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix}.$$

Let $E = \frac{1}{\sqrt{k}} AR$ and let $f : \mathbb{R}^d \to \mathbb{R}^k$ map the i-th row of A to the i-th row of E.
Then

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$$

holds for all $u, v \in P$ with probability at least $1 - n^{-\beta}$.

Implementation – Random projections

Random projections

Corollary of the Johnson–Lindenstrauss lemma:

Theorem 1

Let P be an arbitrary set of n points in $\mathbb{R}^d$, represented as an $n \times d$ matrix $A \in \mathbb{R}^{n \times d}$. Given $\varepsilon > 0, \beta > 0$ let $k_0 = \frac{4+2\beta}{\varepsilon^2/2-\varepsilon^3/3} \log n$. For integer $k \geq k_0$, let $R \in \mathbb{R}^{d \times k}$ be a random matrix with elements r_{ij}, where r_{ij} are independent random variables from either one of the following two probability distributions:

$$r_{ij} : \begin{pmatrix} 1 & -1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad r_{ij} : \begin{pmatrix} 1 & 0 & -1 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix}.$$

Let $\boxed{E = \frac{1}{\sqrt{k}} AR}$ *and let $f : \mathbb{R}^d \to \mathbb{R}^k$ map the i-th row of A to the i-th row of E. Then*

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$$

holds for all $u, v \in P$ with probability at least $1 - n^{-\beta}$.