UNIVERSITÄT
BERN

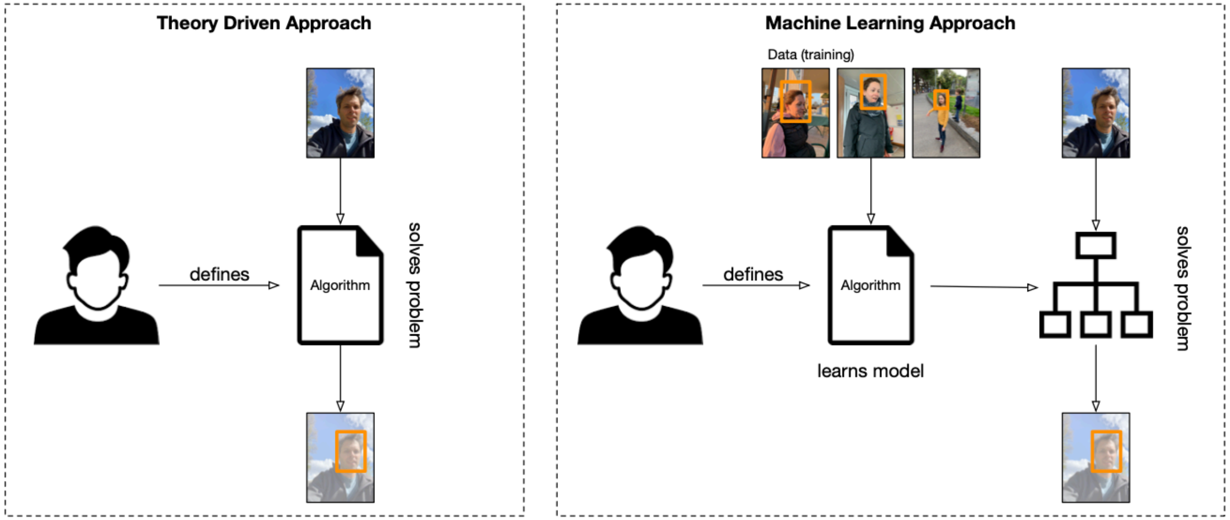
# Graph Based Pattern Recognition

## Repetition of Chapter 1: Introduction

PD Dr. Kaspar Riesen

- *Pattern recognition* is a computer science discipline that aims at defining algorithms that automate and/or support the human process of perception and intelligence.

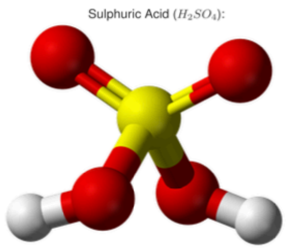
- Pattern recognition employs *machine learning* to learn models from data.



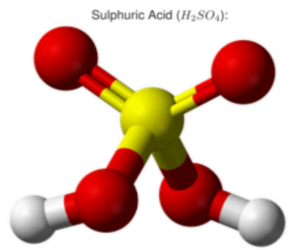

- We distinguish between *supervised* and *unsupervised learning*.

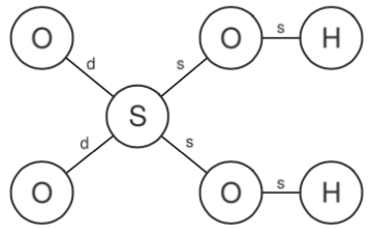
- The question how to represent the underlying data in a formal way is a key issue.

- We distinguish between *statistical and structural representations.*




$$\mathbf{x} = (\underbrace{2, 4, 0, 1}_{\text{Frequency of atoms (H, O, C, S)}}, \underbrace{16, 5.91, \ldots}_{\text{Topological descriptors}}) \in \mathbb{R}^n$$



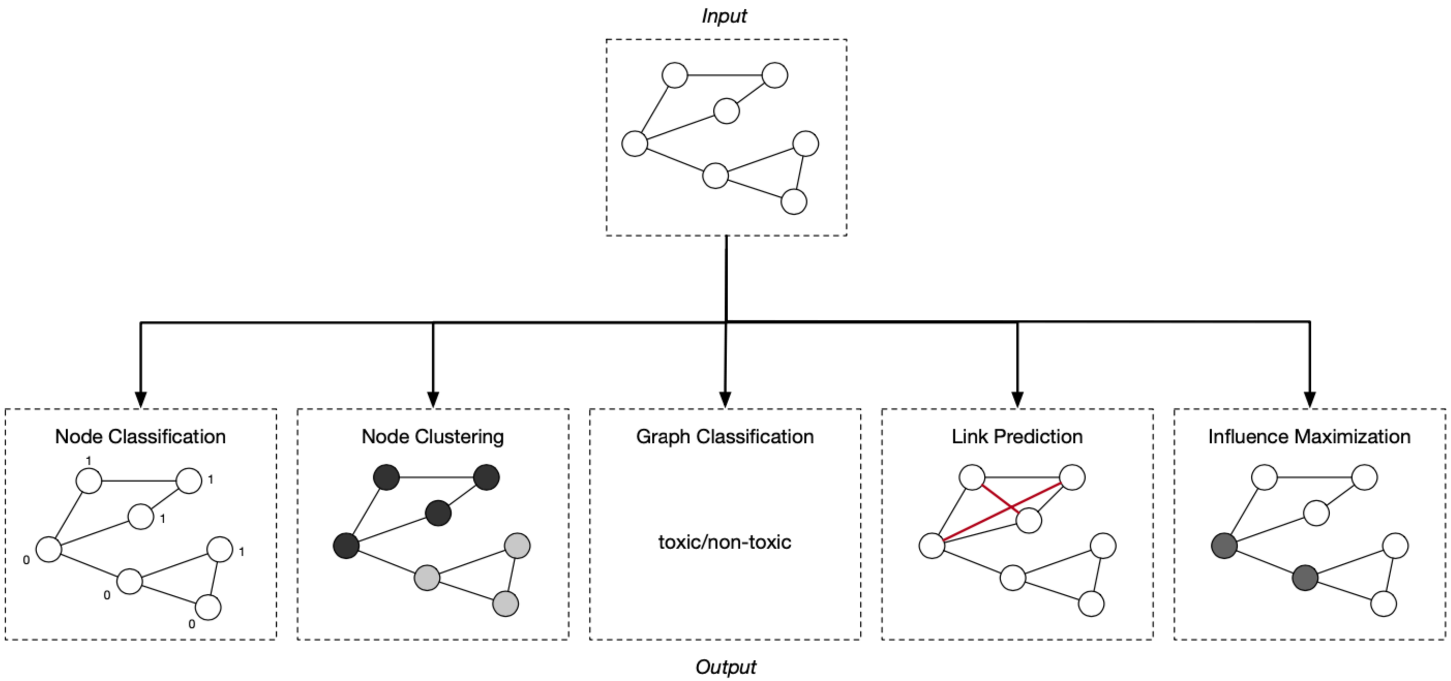




- We observe complementary properties of both approaches:

| | **Vectors** | **Graphs** |
|---|---|---|
| **Representational Power** | Low | High |
| **Efficiency** | High | Low |

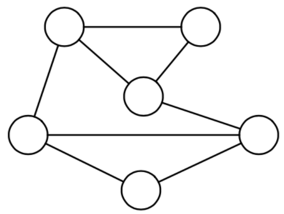
- The field of graph-based pattern recognition has a long tradition and can roughly be subdivided into three main eras:
  - First era: Graph matching and graph clustering
  - Second era: Graph kernels
  - Third era: Graph neural networks
- The present lecture is structured along these three eras (from Chapter 2 to 12).

- In graph based pattern recognition one distinguishes between *graph-level* as well as *node*- and *edge-level tasks:*

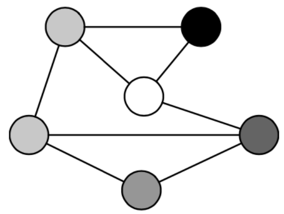


**Definition 2 (Graph)** *Let $L_V$ and $L_E$ be finite or infinite label sets for nodes and edges, respectively. A graph $g$ is a four-tuple $g = (V, E, \mu, \nu)$, where*

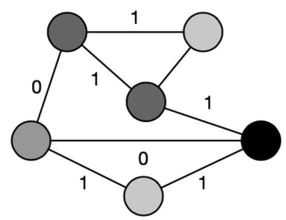
- *$V$ is the finite set of nodes,*
- *$E \subseteq V \times V$ is the set of edges,*
- *$\mu : V \to L_V$ is the node labeling function, and*
- *$\nu : E \to L_E$ is the edge labeling function.*



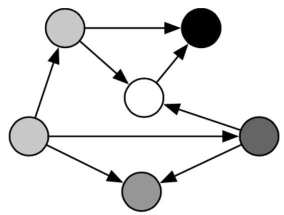
(a)



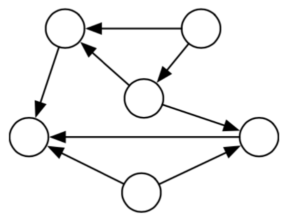
(b)



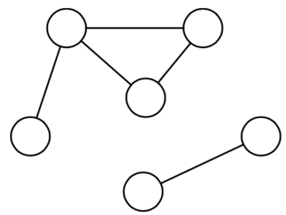

(c)



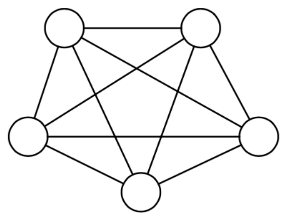
(d)



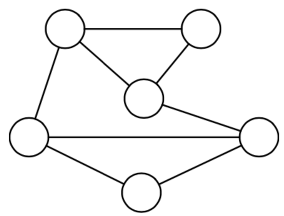
(e)



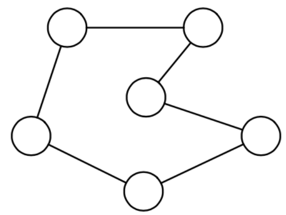
(f)



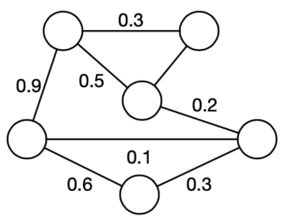
(g)



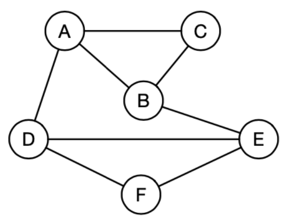
(h)



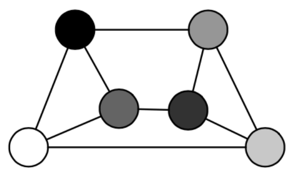
(i)



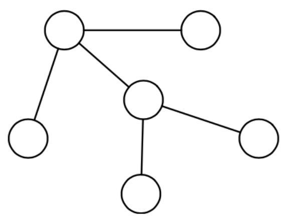

(j)



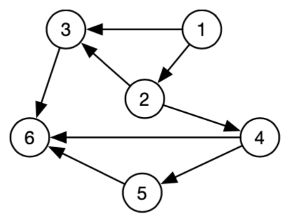

(k)



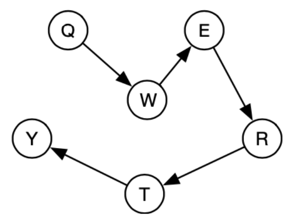
(l)



(m)



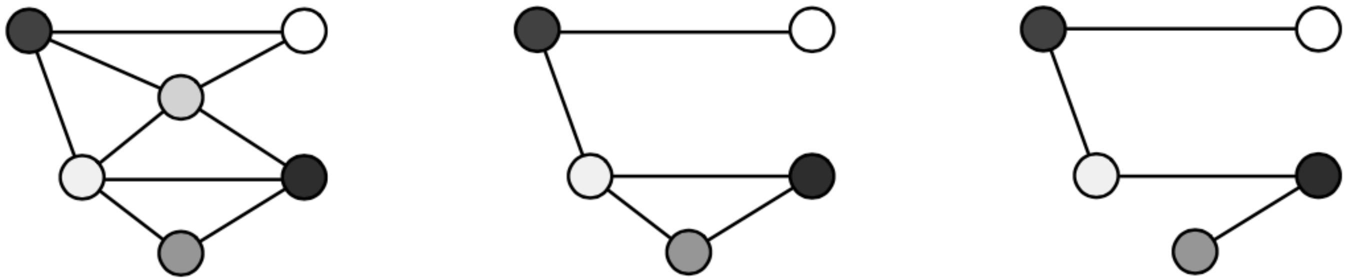

(n)




(o)

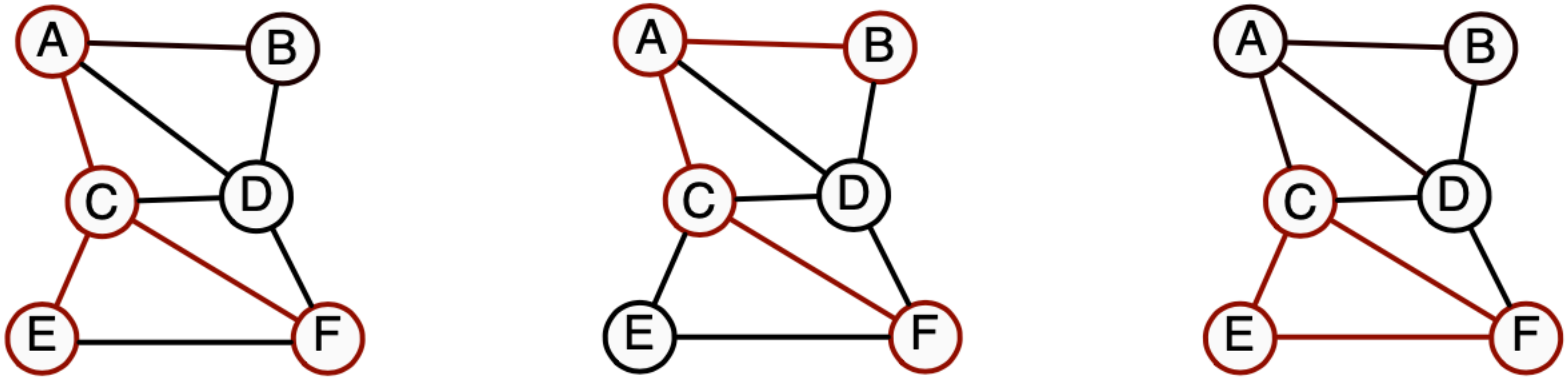
- A *subgraph* $g_1$ is obtained from a graph $g_2$ by removing some nodes and their incident (as well as possibly some additional) edges from $g_2$.

- For $g_1$ to be an *induced* subgraph of $g_2$, some nodes including their incident edges are removed from $g_2$ only, i.e. no additional edge removal is allowed.



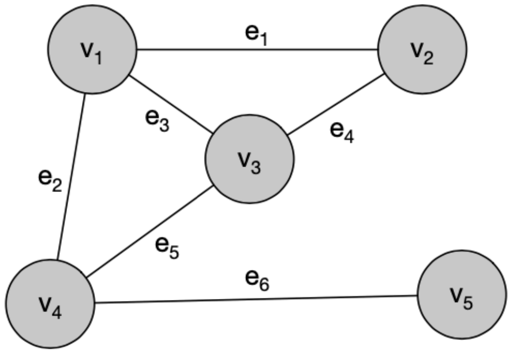
(a) (b) (c)

| Symbol | Meaning |
|---|---|
| $g$ | Graph |
| $V$ | Set of nodes |
| $E$ | Set of edges |
| $(u, v)$ | Edge from source $u$ to target $v$ |
| $u \sim v$ | $u$ and $v$ are adjacent |
| $\mu$, $\nu$ | Node and edge labeling function |
| $L_V$, $L_E$ | Alphabets for node and edge labels |
| $\mathcal{G}$ | Graph domain |
| $\|V\|$ | Graph size |
| $\varepsilon$ | empty node/edge |
| $\mathcal{N}(v)$ | Neighborhood of node $v$ |
| $deg(v)$ | Degree of node $v$ |
| $in(v)$ | In-degree of node $v$ |
| $out(v)$ | Out-degree of node $v$ |
| $\delta(g)$ | Density of graph $g$ |




(a) (b) (c)

- A common approach to describe the edge structure of a graph is to define *structural matrices*:




$$\mathbf{D} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

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