

---

# Recent Approaches to Pattern Discovery and Ranking with Graphs

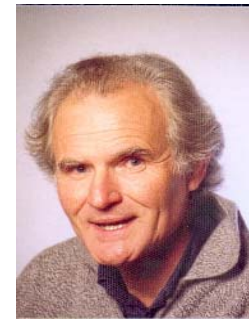
Prof. Dr. Stefan Wrobel

Fraunhofer IAIS and University of Bonn

---

Joint work with

Mario Boley, Dr. Thomas Gärtner, Dr. Tamas Horváth, Dr. Axel Poigné, and Shankar Vembu



# Knowledge discovery and machine learning group

The group is carried by two institutions (CS department University of Bonn and Fraunhofer IAIS, joint appointment of Stefan Wrobel)



Research topics:

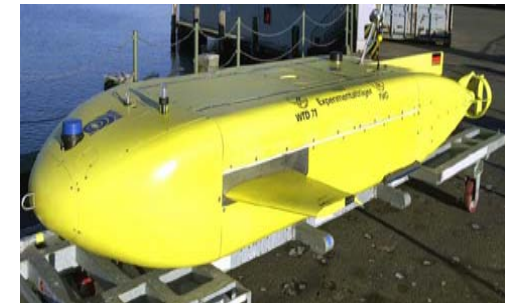
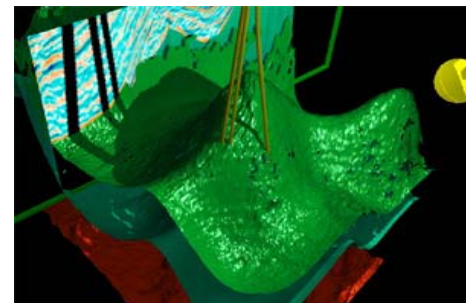
- Predictive and descriptive graph mining
- Semi-supervised regression and ranking
- Transduction on extremely large databases
- Text mining
- Spatial data mining

(Further topics at IAIS)

# Fraunhofer IAIS: Intelligent Analysis and Information Systems

## Core research areas:

- Machine learning and adaptive systems
- Data Mining and Business Intelligence
- Automated media analysis
- Interactive access and exploration
- Autonomous systems



# Outline

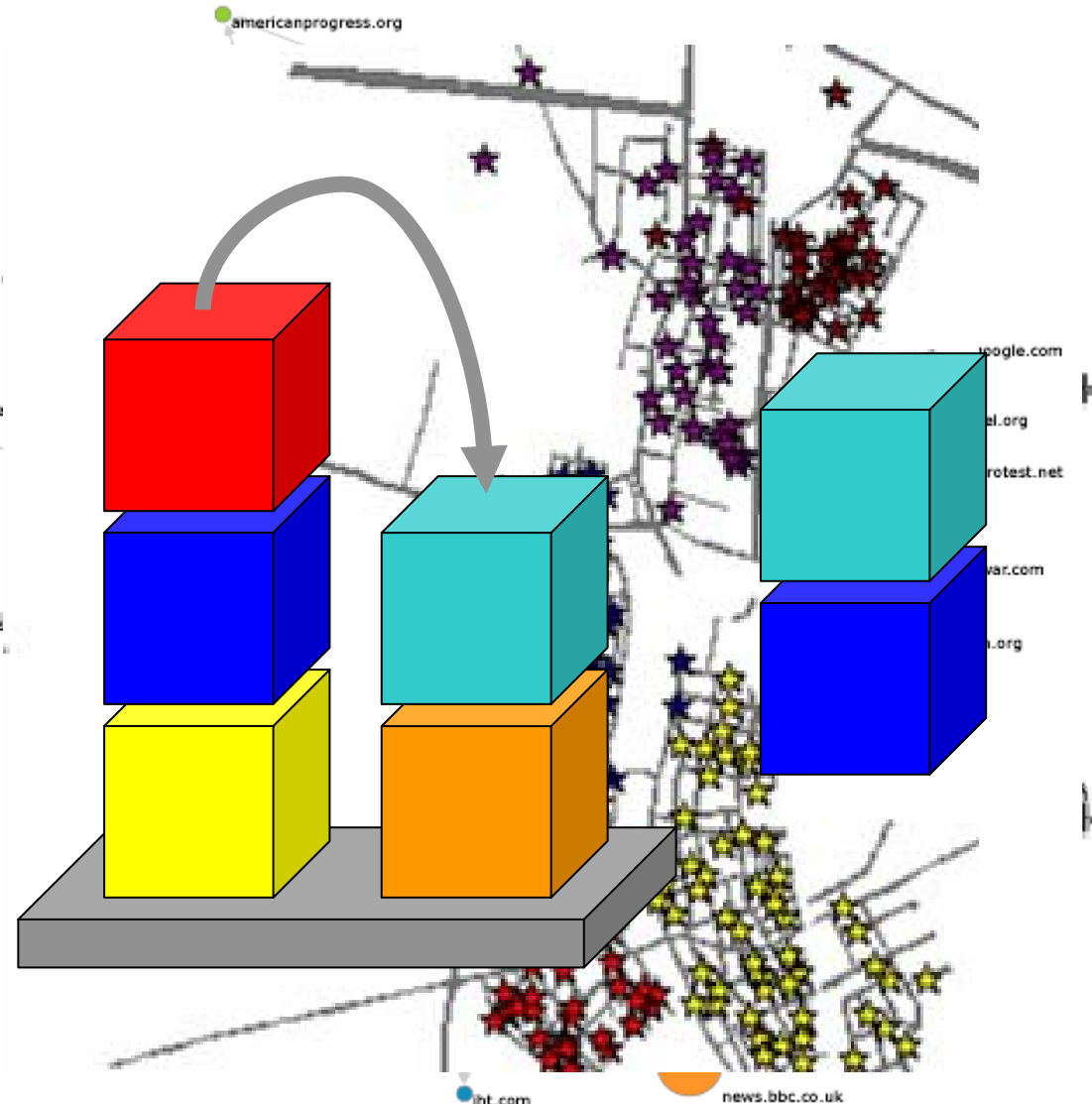
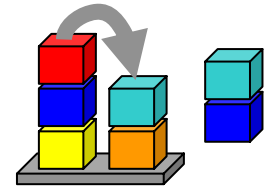
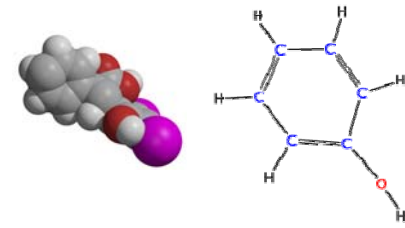
- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

# Outline

- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

# Graph Mining problems abound ...

- mole
- classic
- inade



1234	asdf
477	ab
1234	asdf
477	ab
1234	asdf

Graph I

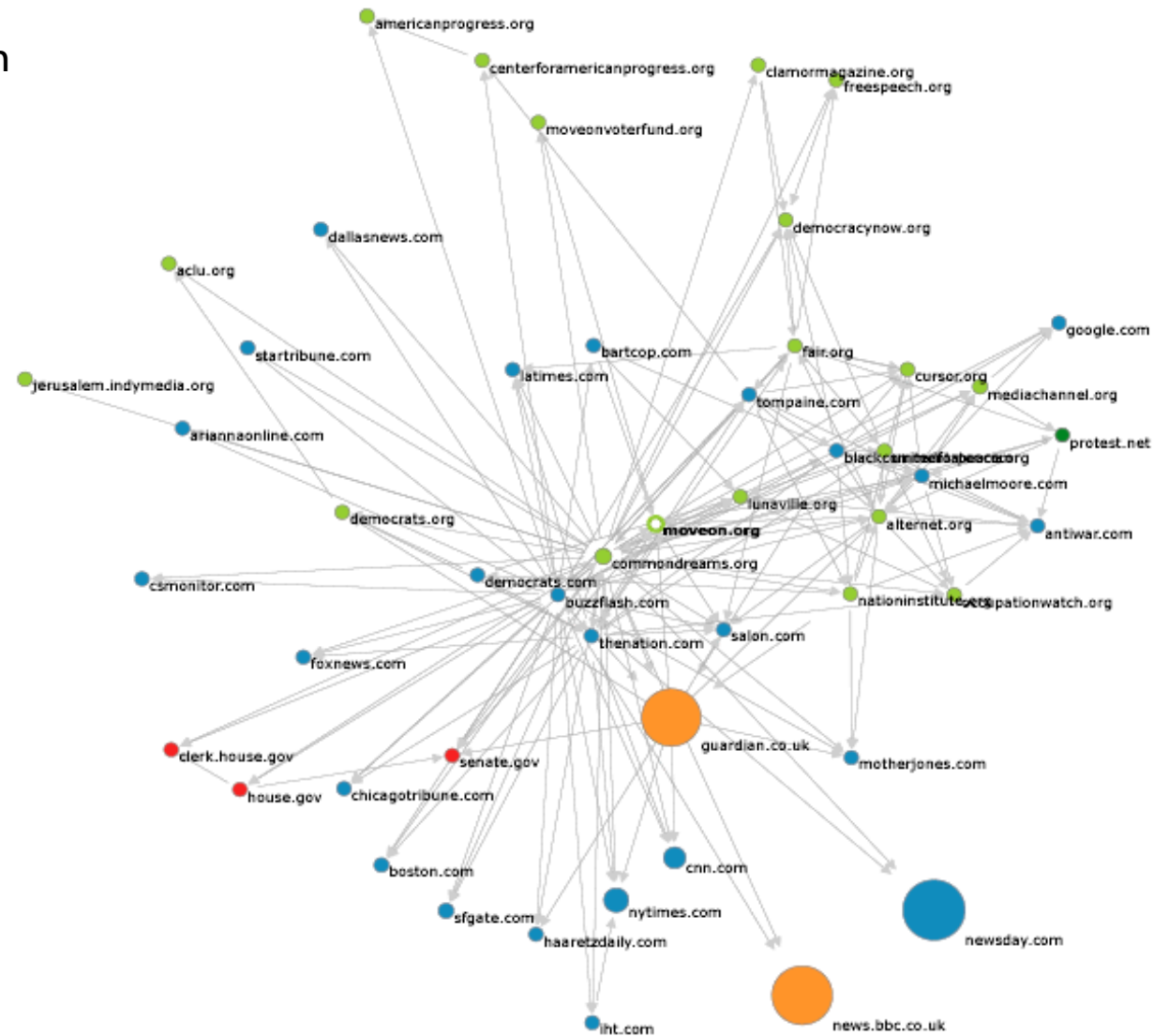
with graphs directly

Wrobel

6

# Classifying documents and websites

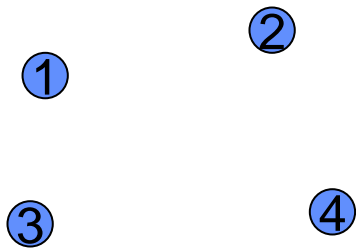
- and the web is a graph



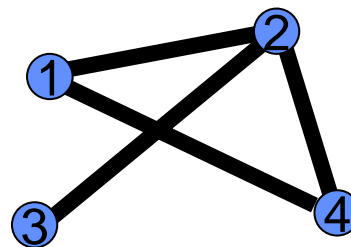
**Farrall, 05**

# Labelled Undirected Graphs

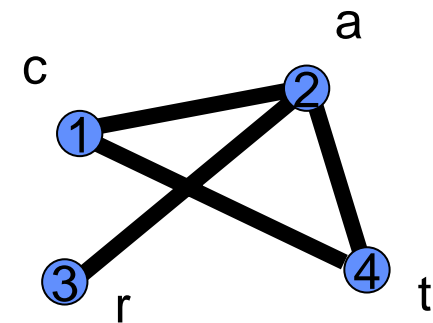
■ vertices



■ edges



■ labels



$$\mathcal{V} = \{\nu_1, \nu_2, \nu_3, \nu_4\}$$

$$\mathcal{E} \subseteq \{e \subseteq \mathcal{V} : |e| = 2\}$$

$$label : \mathcal{V} \cup \mathcal{E} \rightarrow \mathcal{L}$$

$$\mathcal{E} = \{\{\nu_1, \nu_2\}, \{\nu_2, \nu_4\}, \{\nu_4, \nu_1\}, \{\nu_2, \nu_3\}\}$$



# Graph Mining Tasks

- Classification and regression
  - Given a graph or a node, classify into one of several classes, or predict a numerical property
  
- Frequent subgraph mining
  - Output all subgraphs that occur often enough in a given graph database
  
- ranking
  - Given samples from a preference relation and a similarity graph, predict the complete ranking or preference relation

# Outline

- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

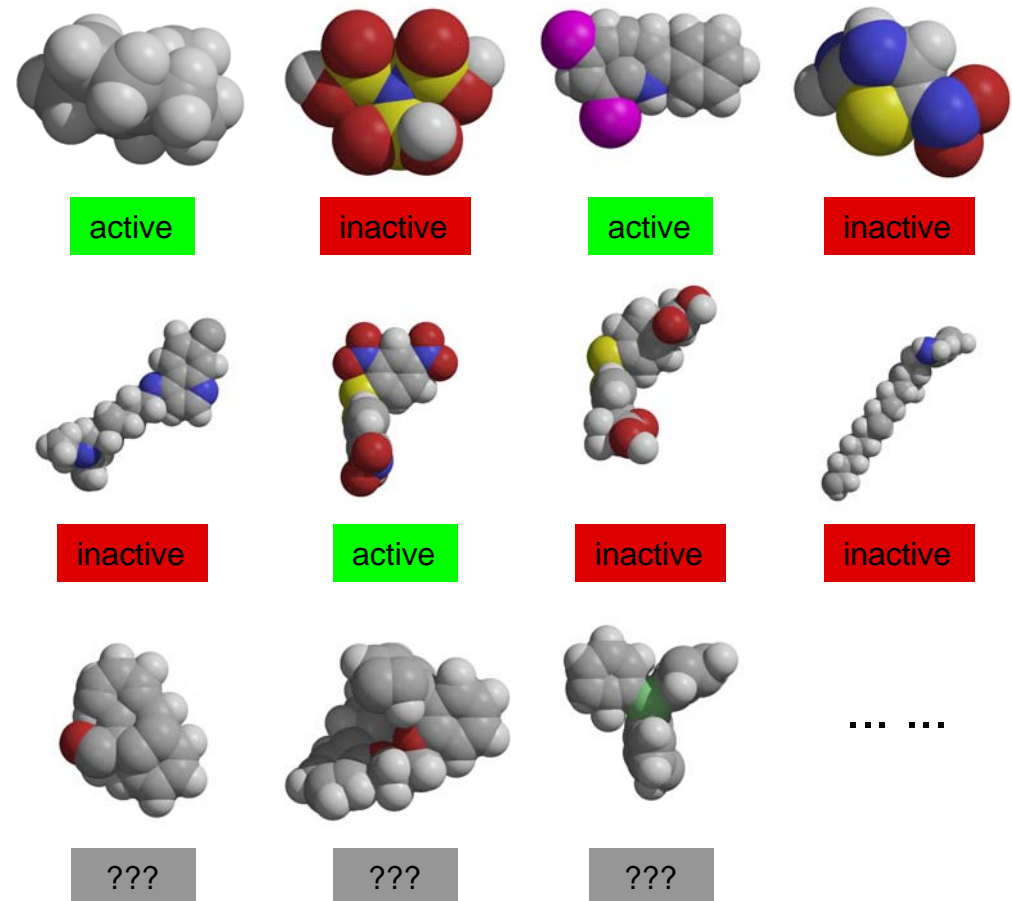
# Scenario: Instances are Labeled Graphs

- many real-world machine learning/data mining problems

Example: [drug screening](#)

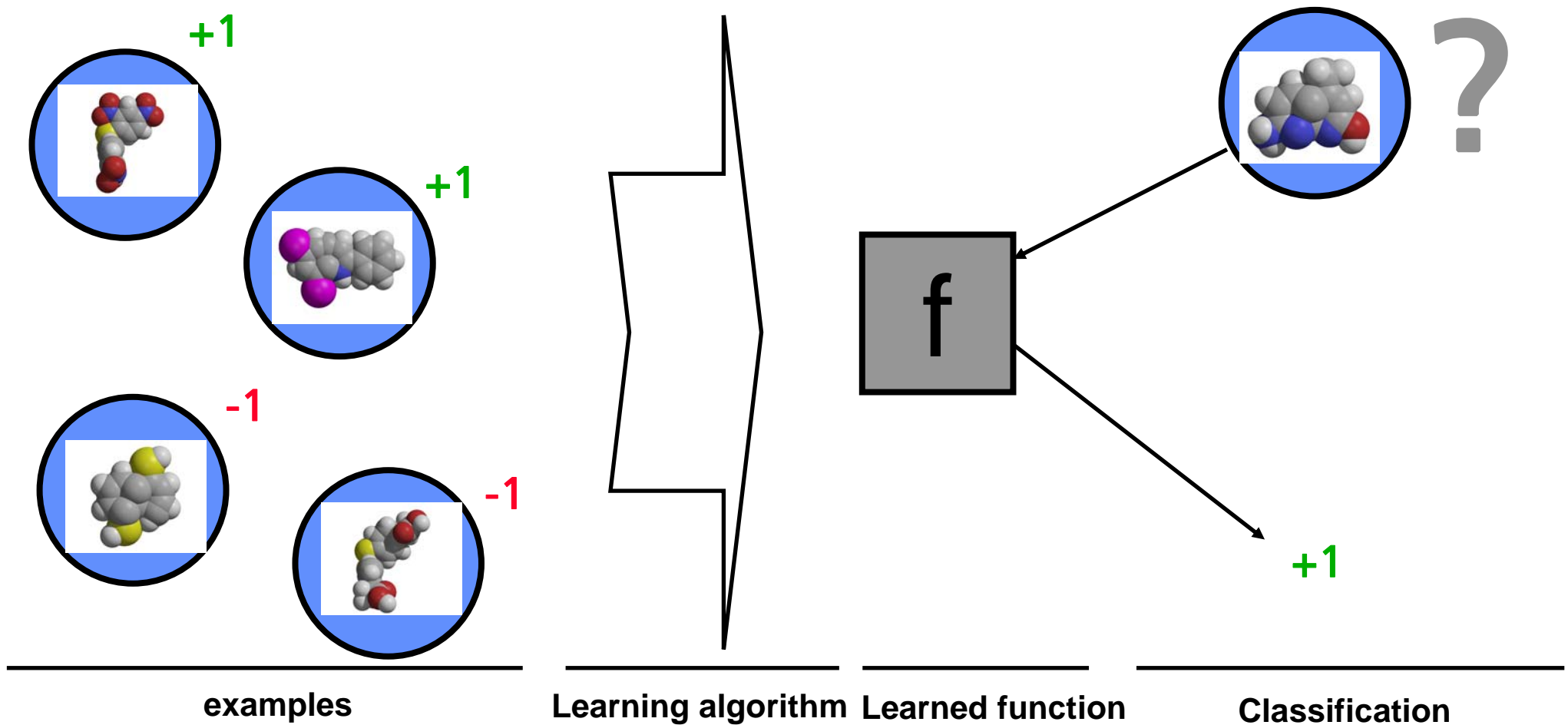
Goal:

- find molecules that are **active** against some disease (e.g., HIV)
- learn to recognize new candidates based on lab-examined examples

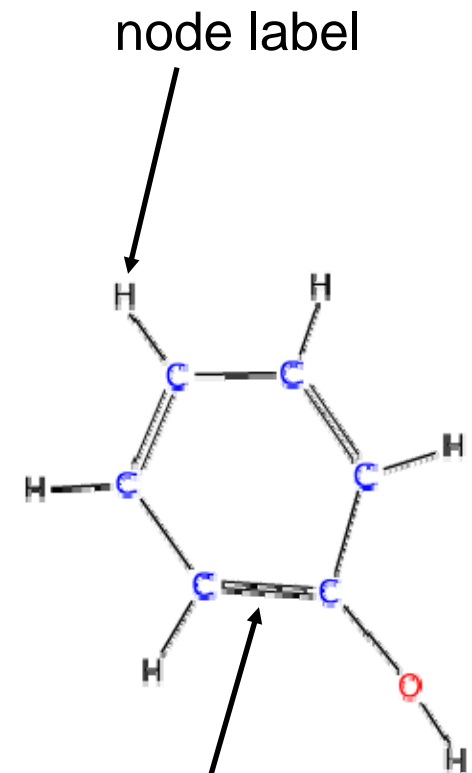
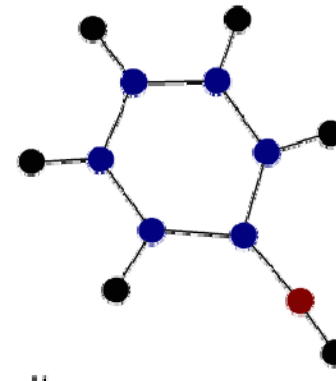
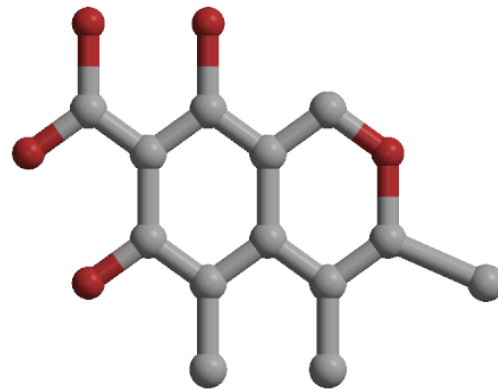
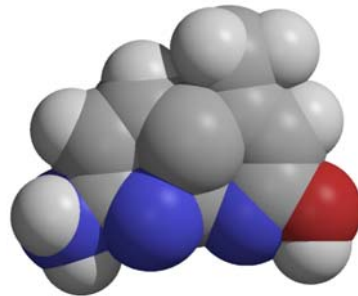
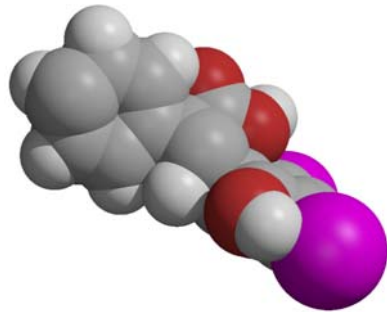


# Predictive Learning from Examples

Goal: learn  $f$  with minimal expected error



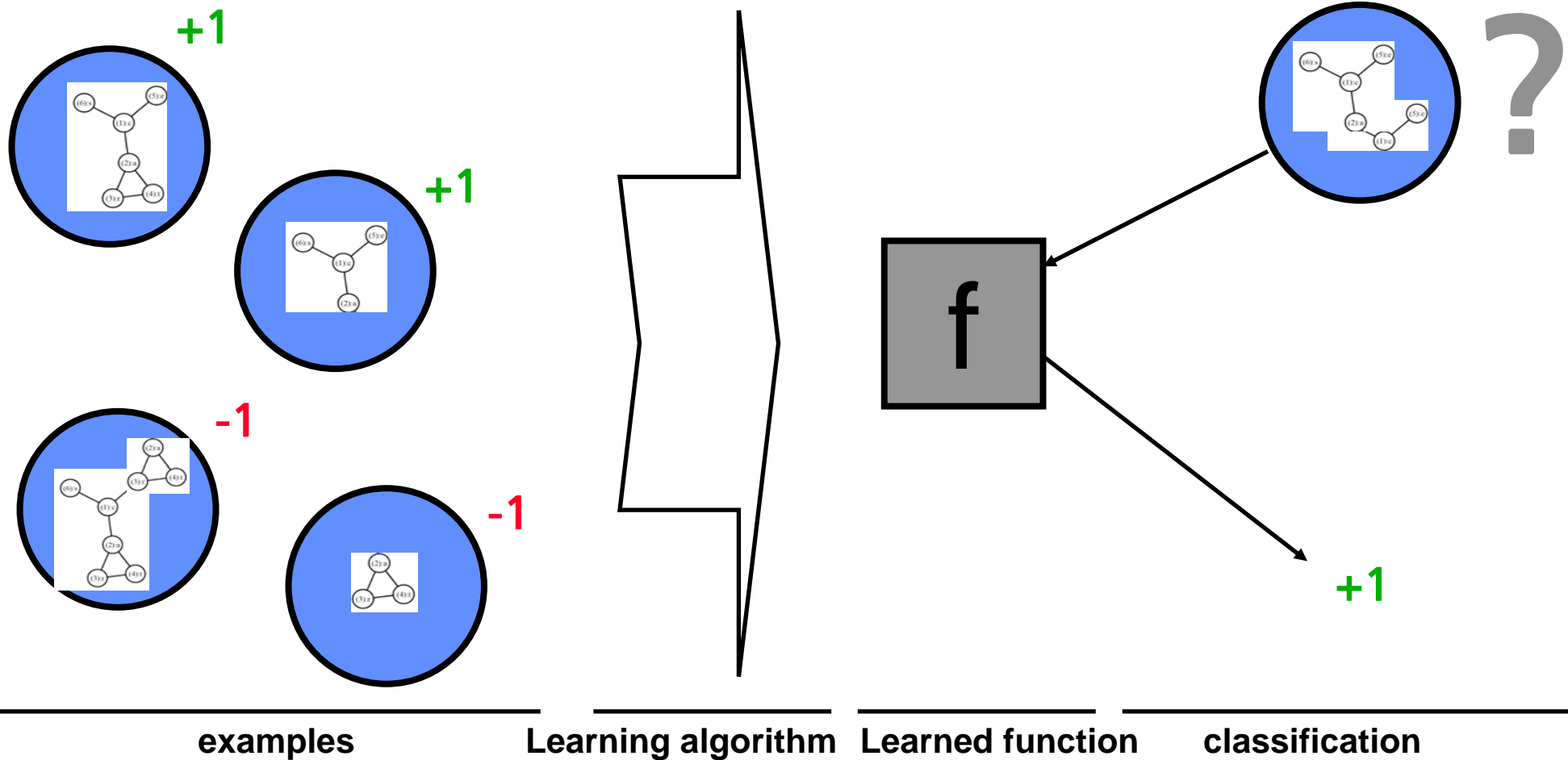
# Molecules and their molecule graphs



Molecules give rise to labelled undirected graphs

# Predictive Learning from Examples

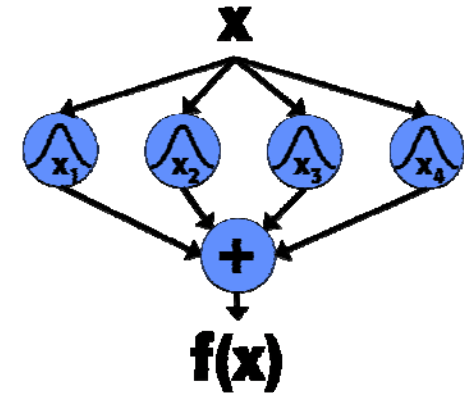
Goal: learn  $f$  with minimal expected error



## Kernel Methods

find linear combination of basis functions

$$f(\cdot) = \sum c_i k(x_i, \cdot)$$

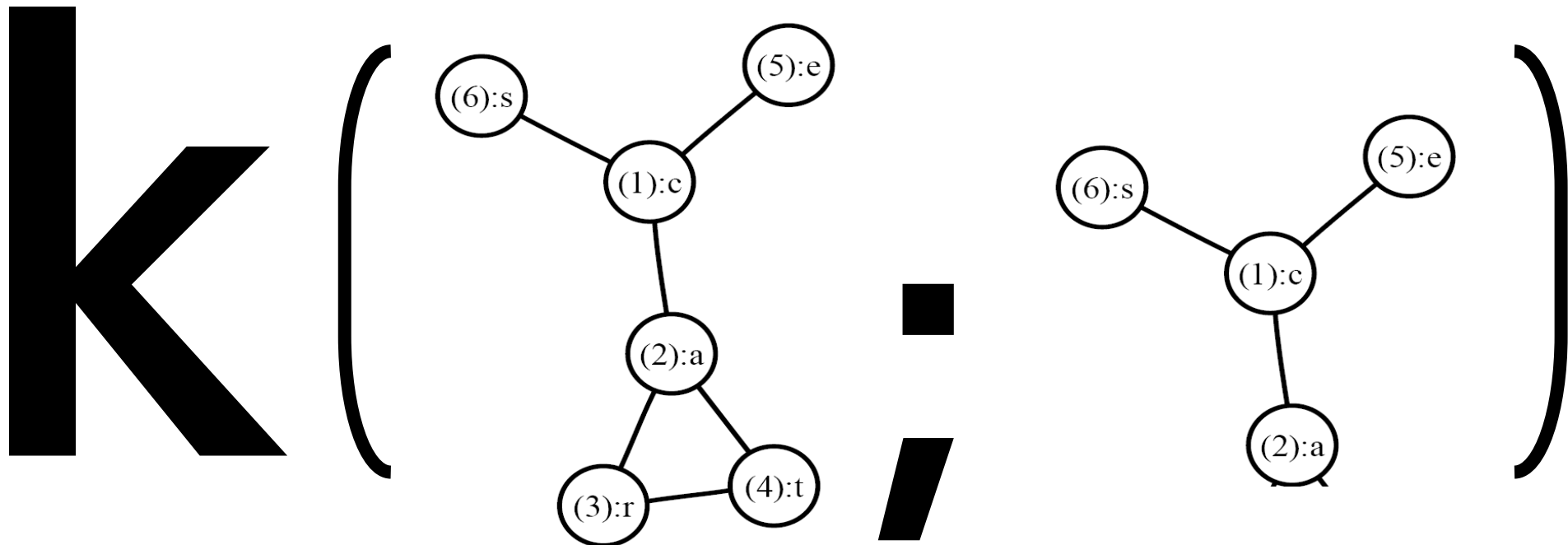


with positive definite  $k : \sum c_i c_j k(x_i, x_j) \geq 0$

- convex optimisation problem / often analytical solutions exist
- geometric interpretation  $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$
- training algorithms for many learning tasks with good mathematical foundation (e.g. support vector machine SVM)
- kernel functions can be designed independent of algorithm, combined easily and can encode domain knowledge if desired

# Kernel for labeled undirected graphs!

positive definite function  $k: \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$



# Graph Kernels

$$\kappa : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{R}$$

Desired properties of **practically useful** graph kernels

- i. computable in polynomial time in the size of the graphs
- ii. able to distinguish between nonisomorphic graphs, i.e., the underlying embedding function is injective modulo isomorphism
- iii. Predictive performance for relevant application problems

Thm [Gärtner, Flach, & Wrobel, 2003] :

Computing any graph kernel that distinguishes non-isomorphic graphs is at least as hard as deciding graph isomorphism.

[Details 1](#)

[Details 2](#)

- ☹ graph isomorphism: although probably not NP-complete, it is believed to be not in P
- ⇒ we have to resort to graph kernels where the underlying embedding functions may map some non-isomorphic graphs to the same point

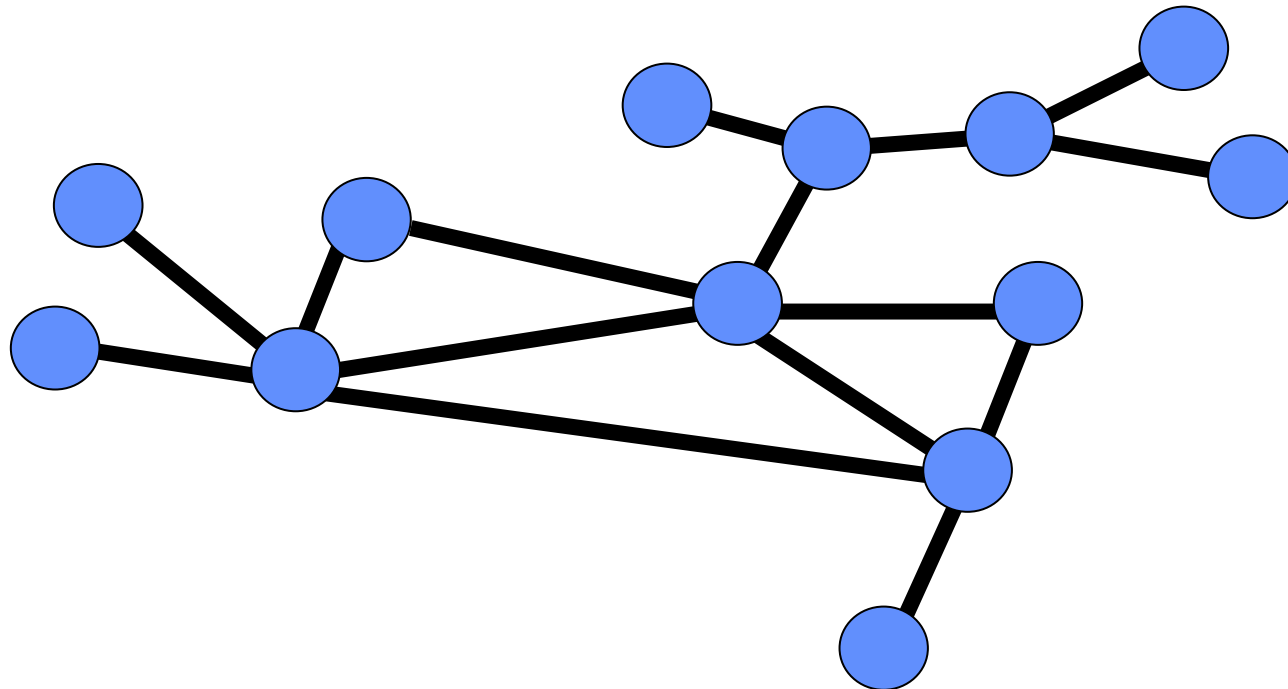
# Graph Kernel Approaches

- consider only some set of subgraphs of given type
  - e.g., only paths or walks (perhaps up to fixed length)
    - kernels based on walks with common label sequences (with and without gaps) can be computed in polynomial time [Gärtner, Flach, Wrobel 03, Gärtner 05]
    - still can't handle 45.000 molecules
  - E.g. cyclic pattern kernels [Horvath, Gärtner, Wrobel/KDD04, Horvath/05]
  
- consider only set of *frequent* subgraphs of given type
  - e.g., frequent subgraphs [Deshpande, Kuramochi, Karypis 03,04]
  - E.g. outerplanar graphs [Horvath, Ramon, Wrobel/KDD06]

# Cyclic pattern kernels

- [Horvath, Gärtner, Wrobel/KDD04, Horvath/PAKDD05]

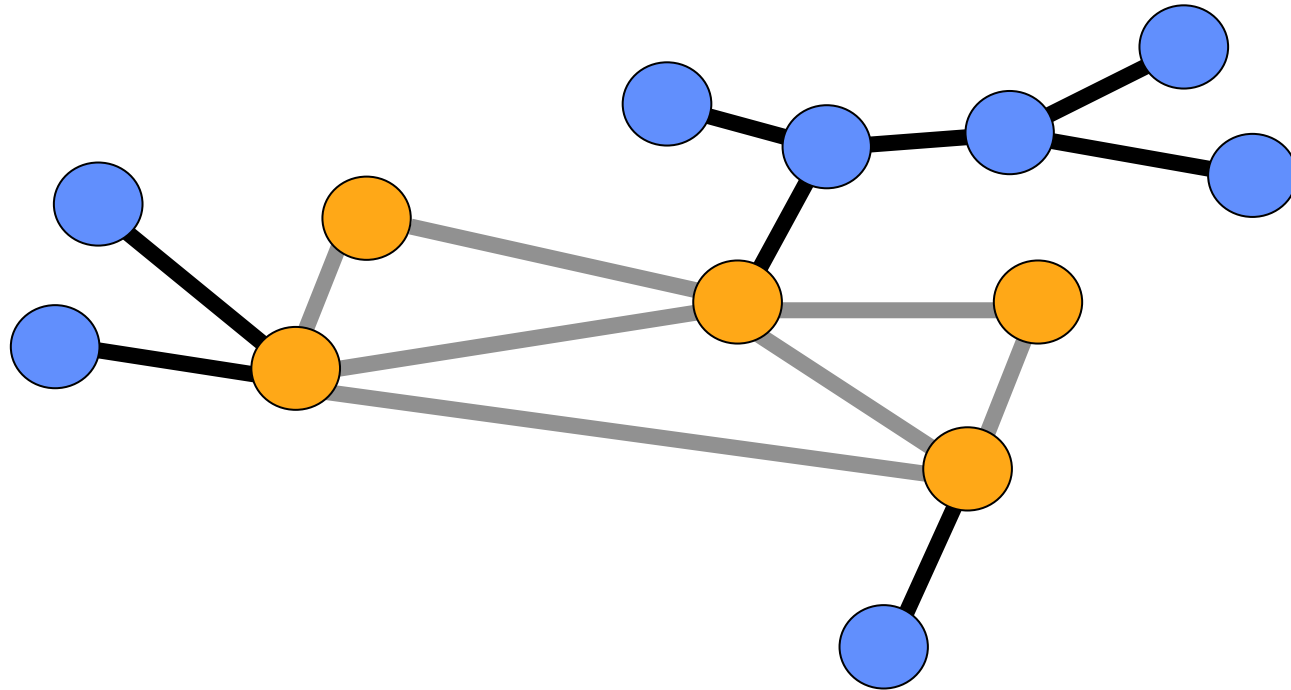
# Cyclic Pattern Kernels [Horvath, Gärtner, Wrobel/KDD04, Horvath/PAKDD05]



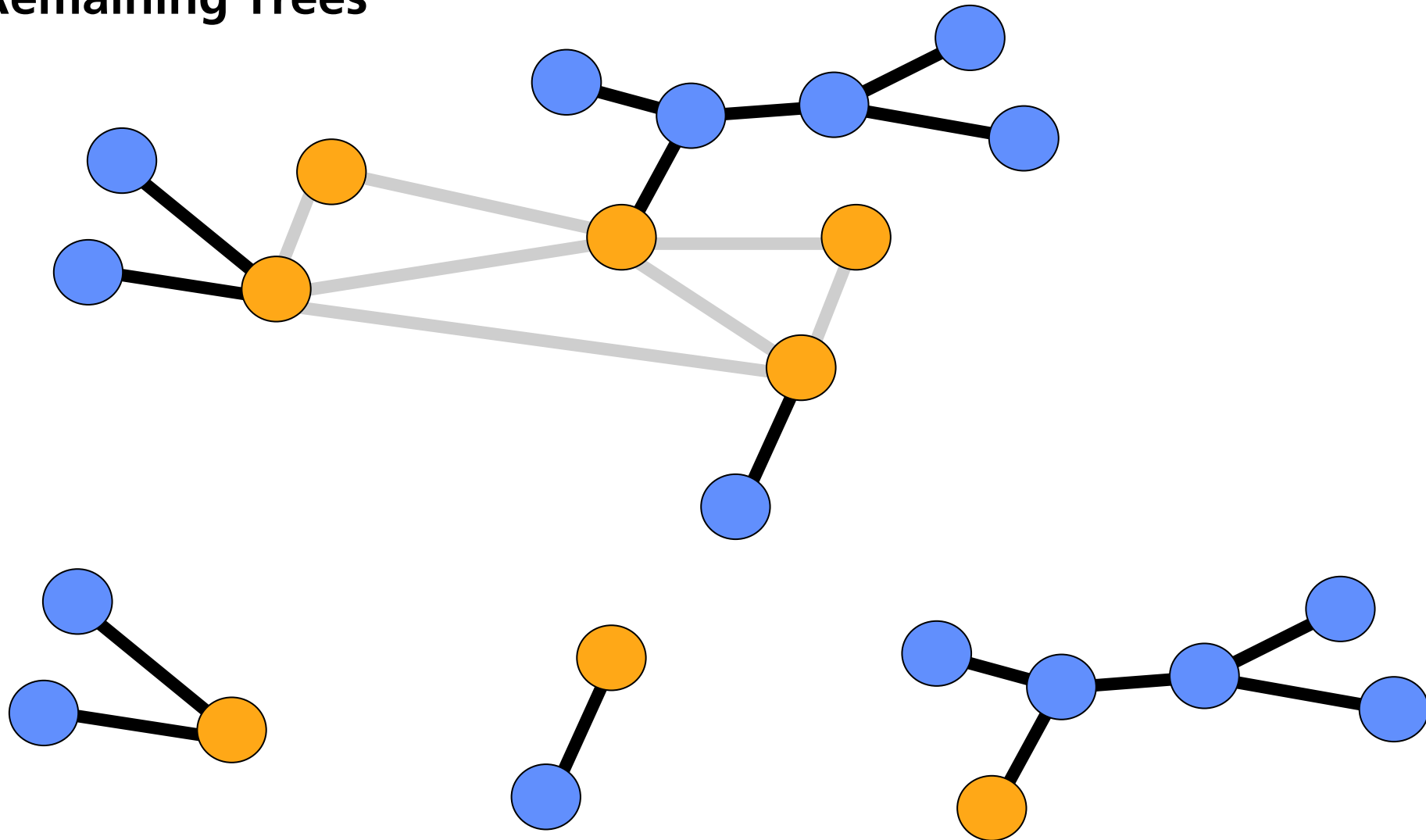
- Handling a graph is difficult if it contains cycles
- we handle the part containing cycles and the part containing trees separately!

Cf. Role of rings in molecules (Peter Ertl, Novartis, 7th ICCS Conference , 2005)

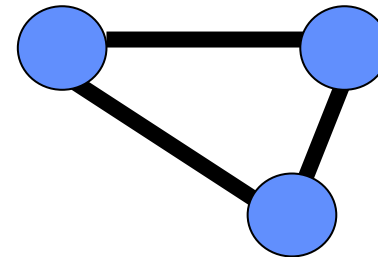
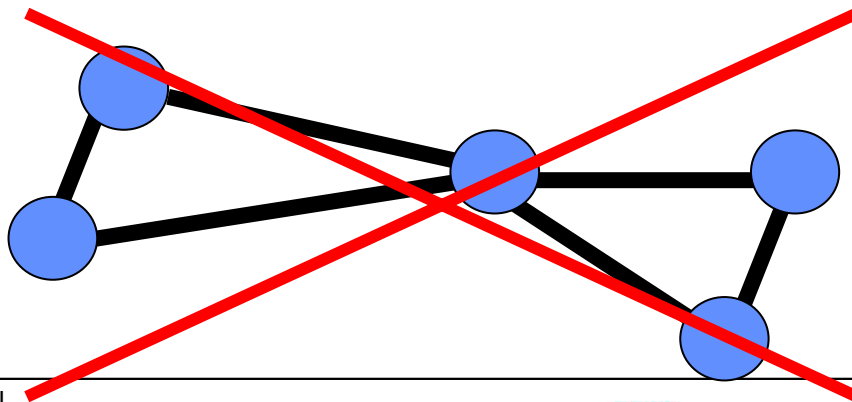
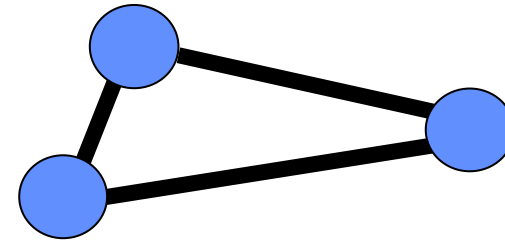
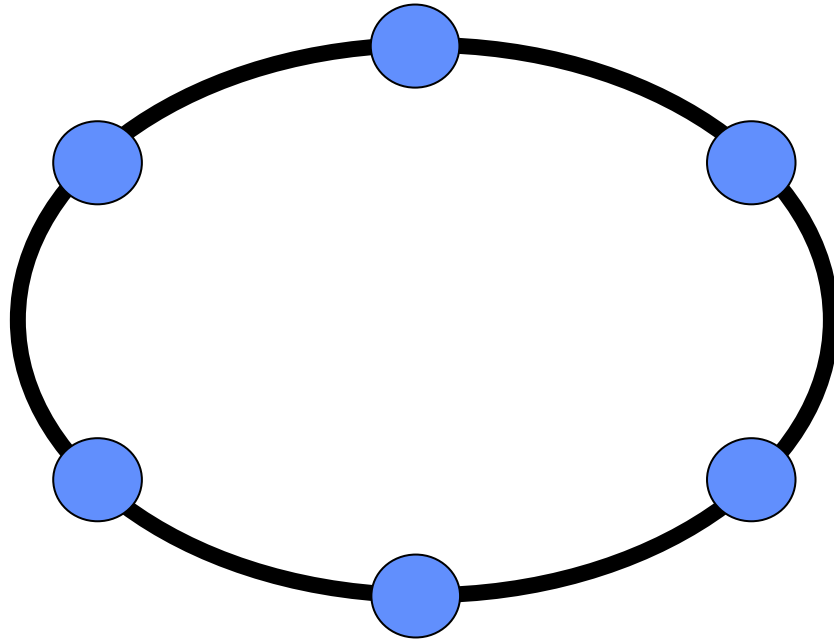
# Biconnected Component



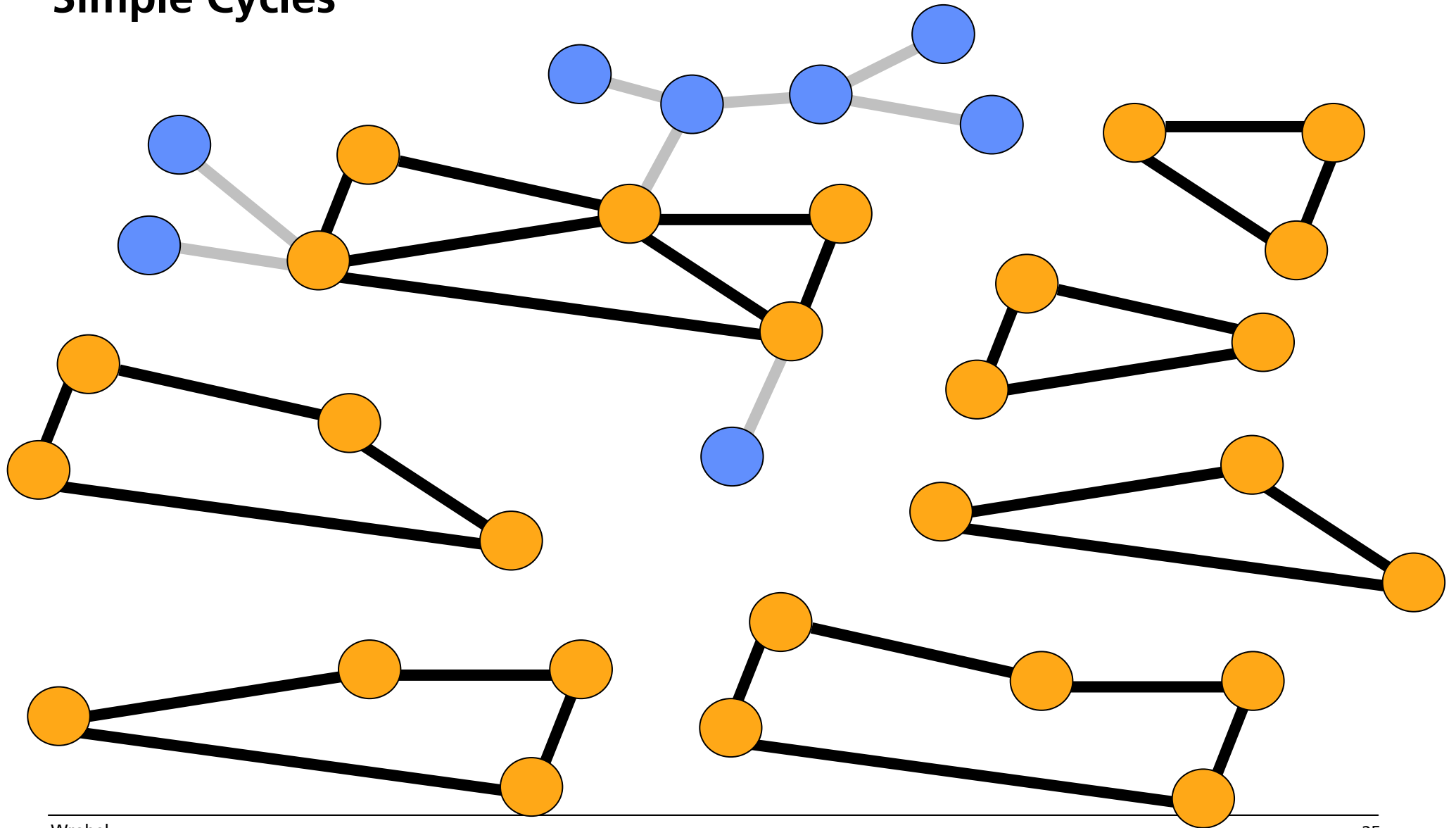
# Remaining Trees



# Simple Cycles



# Simple Cycles



# Notions

## simple cycle:

- connected graph such that each vertex has degree 2
- $\mathcal{S}(G)$ : set of simple cycles of a graph  $G$

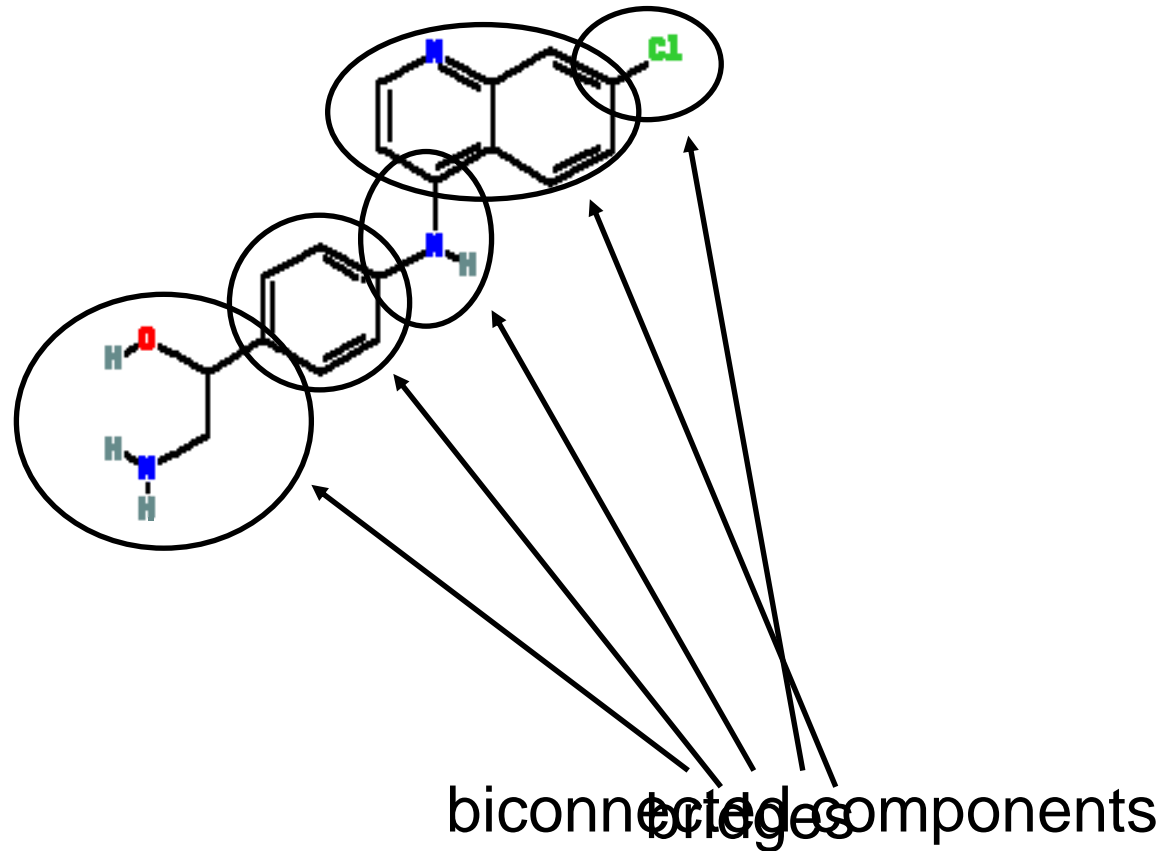
## biconnected component:

- maximal connected subgraph such that each edge belongs to a simple cycle

## bridge:

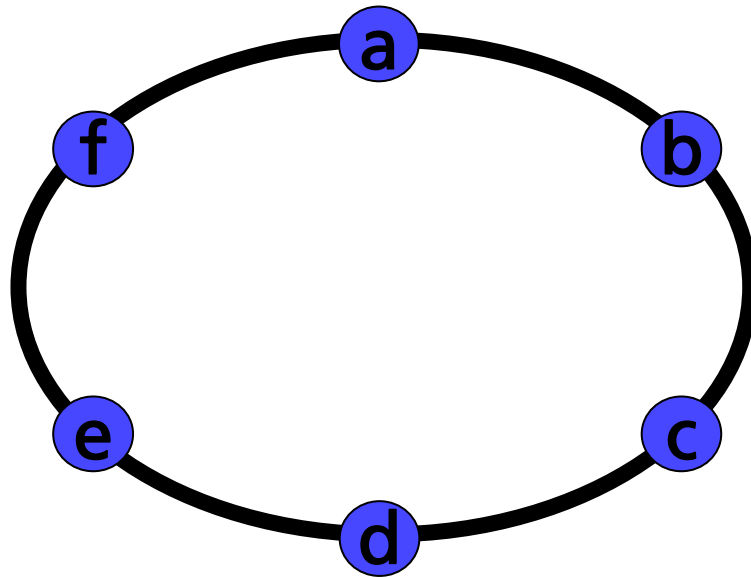
- edge not belonging to simple cycles
- $\mathcal{B}(G)$ : set of bridges of a graph  $G$
- $\mathcal{B}(G)$  is a forest

# Example

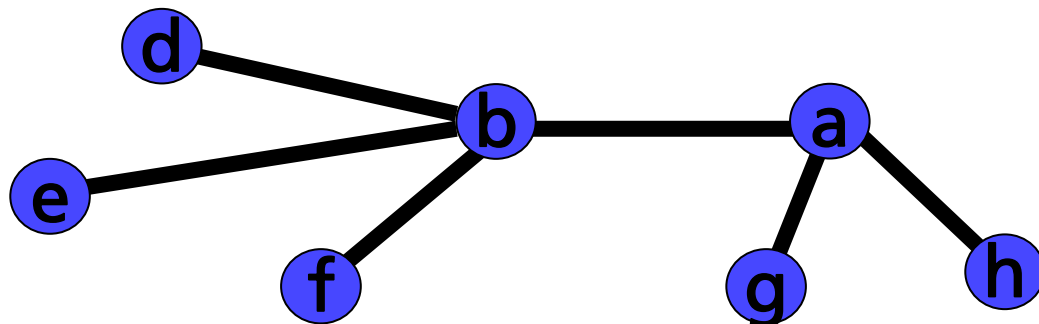


- 2 biconnected components containing 4 simple cycles
- bridges form a forest consisting of 3 disjoint trees

# Cyclic and Tree Patterns

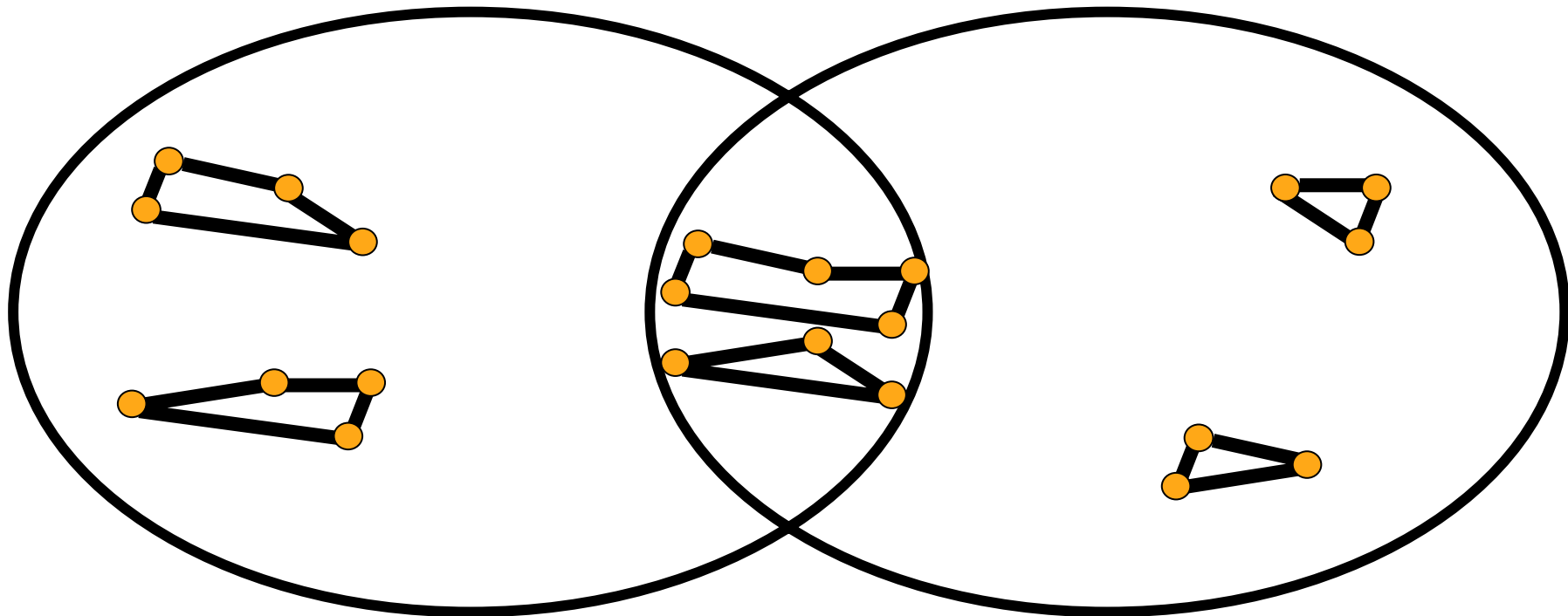


bcdefa  
bafedc  
afdcb  
abcdef



abefdgh  
baghdef  
dbfaghe  
abcdefgh

# Cyclic Pattern Kernels are Intersection Kernels



$$K(X, X') = \mu [X \cap X']$$

## Cyclic Pattern Kernel: Definition

- $G_1, G_2$  : labeled graphs
  - edges and vertices are labeled
- $\pi$ : function mapping the set of simple cycles and trees to strings such that  $\pi$  is injective modulo isomorphism

Cyclic Pattern Kernel (CPK):

$$\kappa_{\mathcal{S}}(G_1, G_2) := |P_{\mathcal{S}}(G_1) \cap P_{\mathcal{S}}(G_2)| + |P_{\mathcal{T}}(G_1) \cap P_{\mathcal{T}}(G_2)|$$

$$P_{\mathcal{S}}(G) := \{\pi(C) : C \in \mathcal{S}(G)\}$$

Example

$$P_{\mathcal{T}}(G) := \{\pi(T) : T \text{ is a tree of } \mathcal{B}(G)\}$$

# Algorithm for Computing CPK

Given labeled graphs  $G_1$  and  $G_2$ :

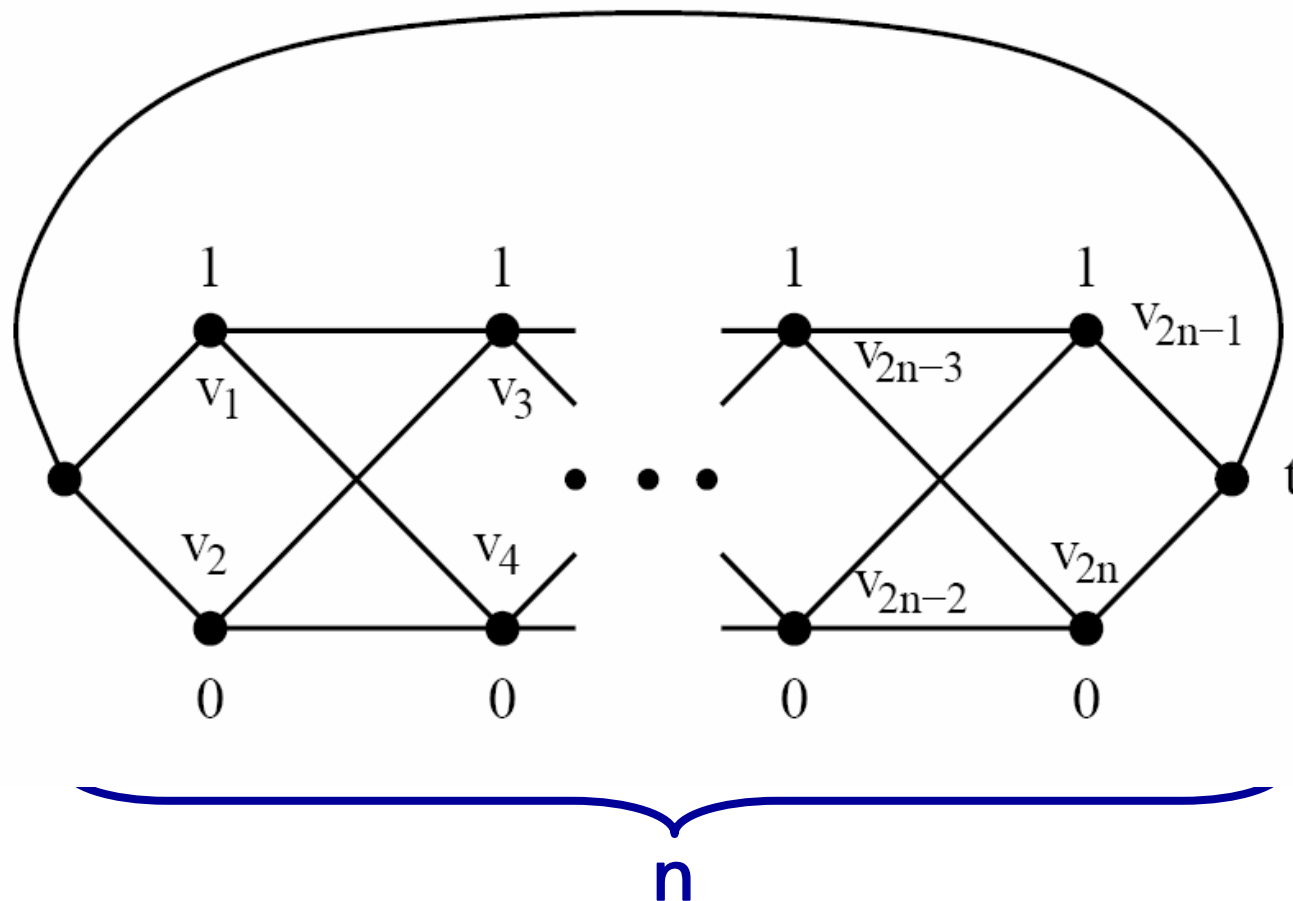
1. compute their biconnected components
  - can be solved in linear time [pseudocode](#)
    - (Tarjan,'72)
2. compute their sets of simple cycles
  - simple cycles are enumerable with linear delay
    - (Read & Tarjan,'75)
3. from these sets, compute the sets of cyclic and tree patterns
4. compute the intersections and their cardinalities

# CPK: Properties

1. CPK is a kernel
2. unless  $P = NP$ , CPK cannot be computed in time polynomial in the size of the input graphs  
[Details 1](#)      [Details 2](#)
  - ☹️ hopeless to find a closed form that can be used to calculate CPK efficiently
3. the corresponding embedding function is not injective modulo isomorphism
  - ☹️ in general, if we require the corresponding embedding function of a graph kernel  $k$  to be injective on non-isomorphic graphs then computing  $k$  is as hard as deciding [graph isomorphism](#) (Gärtner, Flach, & Wrobel, 2003)

# Intractability due to simple cycles/cyclic patterns

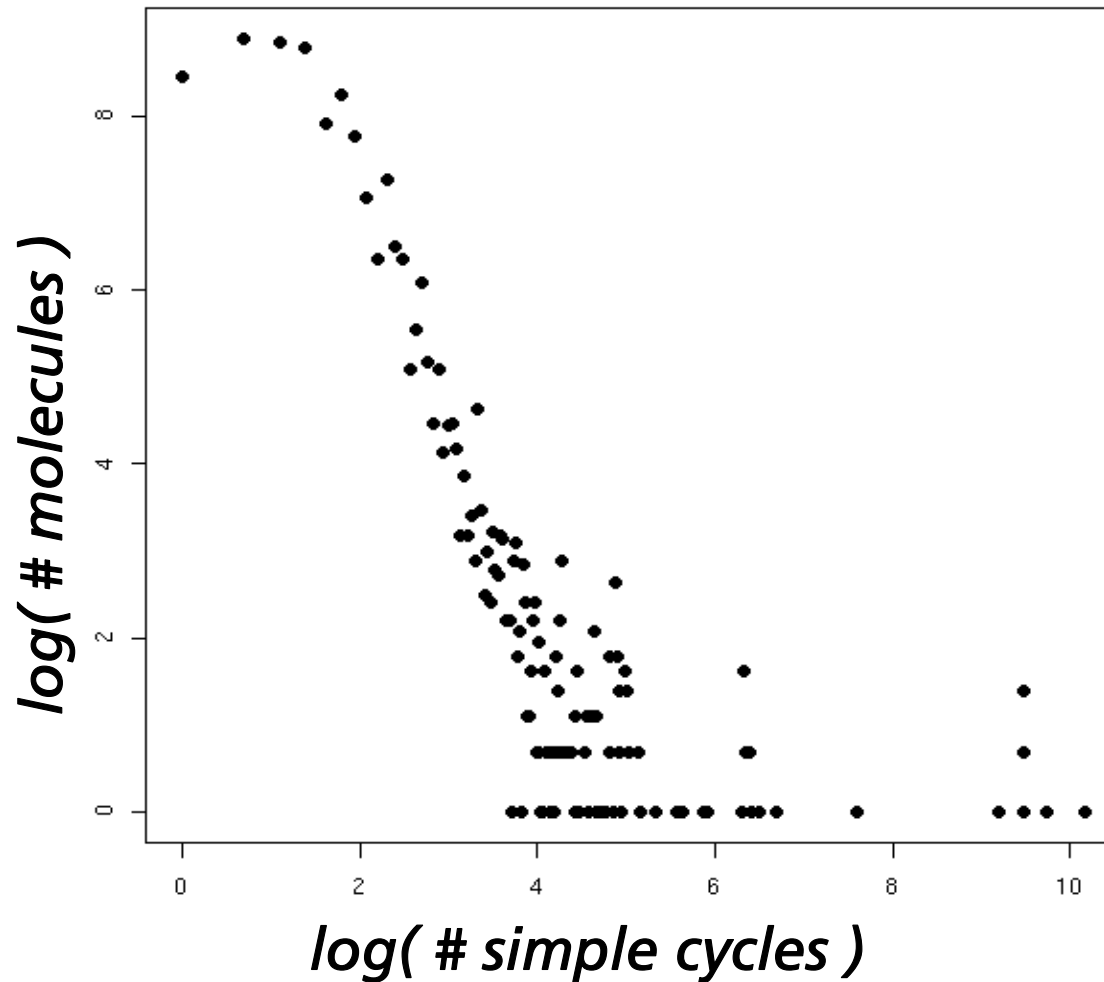
- exponential number of simple cycles / cyclic patterns



## CPK: Scalability Properties

- CPK can be computed in time polynomial in the number of simple cycles of the input graphs
  - ⇒ applicable, if the graphs do not contain too many simple cycles
  - ☺ chemical graphs in pharmacology appear to satisfy this constraint because many of them are d-tenuous outerplanar graphs

# The Real World -- Molecules are 'Well-Behaved'



simple cycles	compounds	fraction
0	1655	3.88%
1 – 9	36026	84.40%
10 – 19	4012	9.40%
20 – 29	514	1.20%
30 – 59	306	0.72%
60 – 99	75	0.18%
100 – 199	68	0.16%
200 – 1000	20	0.05%
> 1000	11	0.03%

# Empirical Evaluation

## Problem:

- find molecules that are active against HIV (AIDS)
  - learn to recognize new candidates based on lab-examined examples

NCI-HIV dataset consisting of **42,689** molecules

- 423 active, 1,081 moderately active, and 41,185 inactive

compared with FSG-approach (frequent patterns) [Deshpande, Kuramochi, and Karypis, 2003]

area under the ROC curve (AUC), more is better

Walk based kernels can handle this problem only up to length 13

## Empirical Results (NCI-HIV dataset)

### Results:

- 3 problems: CA vs. CM, CA+CM vs. CI, CA vs. CI
- time for computing the feature vectors: about 10 minutes
  - Pentium III, 850MHz, SVM<sup>Light</sup> [Joachims, 1999]
- cyclic pattern kernels win very significantly in all 18 problem settings
  - 99% significance level

CA vs CM			CA+CM vs CI			CA vs CI		
cost	$\gamma$ CPK	FSG	cost	$\gamma$ CPK	FSG	cost	$\gamma$ CPK	FSG
1.0	<b>0.8264</b> ( $\pm 0.010$ )	0.7740	1.0	<b>0.8092</b> ( $\pm 0.014$ )	0.7420	1.0	<b>0.9257</b> ( $\pm 0.005$ )	0.8683
1.5	<b>0.8328</b> ( $\pm 0.009$ )	0.7802	1.5	<b>0.8176</b> ( $\pm 0.015$ )	0.7504	1.5	<b>0.9311</b> ( $\pm 0.007$ )	0.8676
2.0	<b>0.8384</b> ( $\pm 0.010$ )	0.7860	15.0	<b>0.8373</b> ( $\pm 0.012$ )	0.7864	15.0	<b>0.9466</b> ( $\pm 0.008$ )	0.9023
2.5	<b>0.8398</b> ( $\pm 0.010$ )	0.7816	35.0	<b>0.8332</b> ( $\pm 0.013$ )	0.7783	35.0	<b>0.9441</b> ( $\pm 0.008$ )	0.9097
3.0	<b>0.8402</b> ( $\pm 0.008$ )	0.7841	50.0	<b>0.8315</b> ( $\pm 0.013$ )	0.7731	50.0	<b>0.9430</b> ( $\pm 0.008$ )	0.9122
15.0	<b>0.8341</b> ( $\pm 0.020$ )	0.7566	100.0	<b>0.8298</b> ( $\pm 0.014$ )	0.7486	100.0	<b>0.9426</b> ( $\pm 0.007$ )	0.9138

# Provably Efficient Cyclic Pattern Kernel variants

- ☹ Thm: Unless  $P = NP$ , CPK cannot be computed in time polynomial in the number of cyclic patterns.

approaches relaxing this computational limitation:

1. for graphs of bounded treewidth:

Thm: CPK can be computed in time polynomial in the number of cyclic patterns

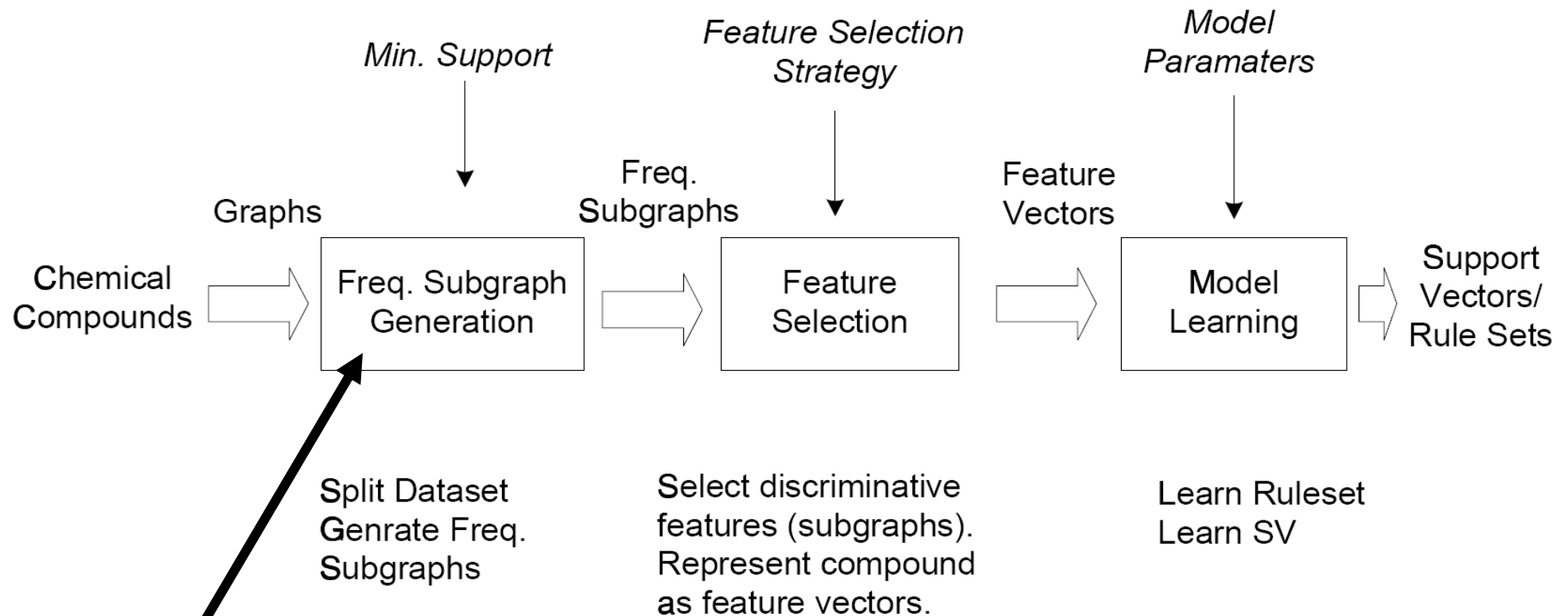
2. predictive performance doesn't decrease for CPK defined by the patterns of relevant cycles

- their number is typically only cubic (Gleiss & Stadler, 1999)
- are enumerable with polynomial delay (Vismara, 1997)
- relevant cycles are efficiently countable (Vismara, 1997)

# Outline

- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

# Predictive graph Mining based on frequent patterns



Consider this subproblem

[Deshpande, Kuramochi, Karypis 03]

# Frequent Subgraph Mining

Given a set  $D$  of labeled graphs and an integer  $t \geq 0$ , enumerate the set of  $t$ -frequent *connected* subgraphs of  $D$  w.r.t. *subgraph isomorphism* ([Details](#))

- ☹ cannot be solved in [output-polynomial time](#) (unless  $P = NP$ )
  - can be used to decide e.g. the Hamiltonian path problem
  - existing approaches resort to various heuristic strategies and restrictions of the search space (often with good empirical performance)
- ☺ solvable with [incremental-polynomial delay](#) if  $D$  is a set of forests
  - [Chi, Muntz, Nijssen & Kok, j05; *survey paper*]

What about problem classes beyond trees?

- **challenge for graph mining:**  
systematic study of graph classes and non-standard matching operators (specialization)

# This Work [Horvath/Ramon/Wrobel/KDD 2006]

## problem class:

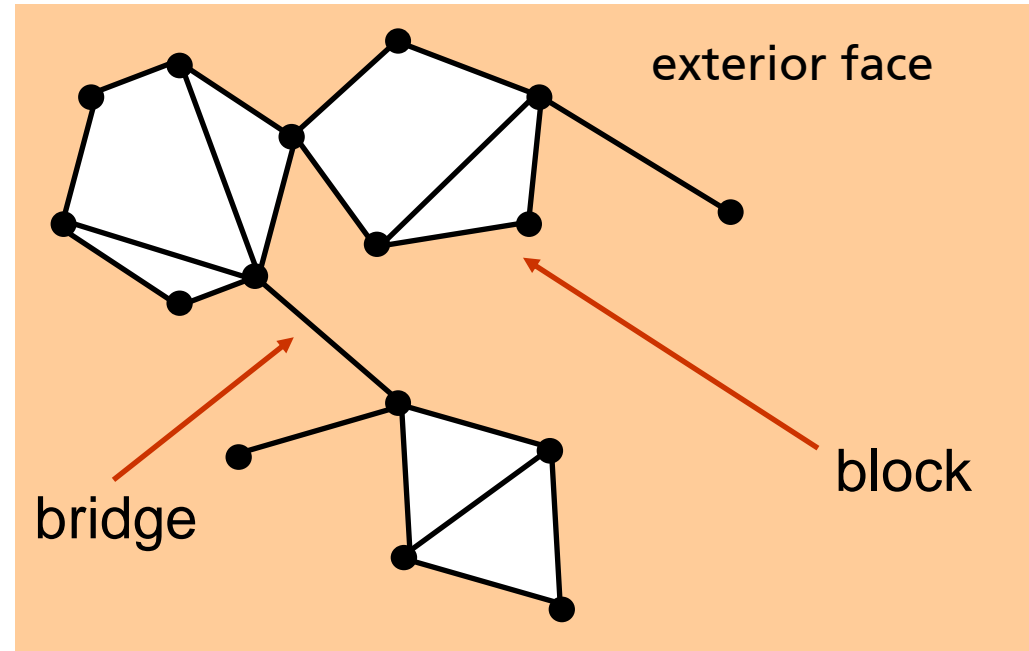
- D: labeled d-tenuous outerplanar graphs
- pattern matching: block and bridge preserving subgraph isomorphism
  - constrained subgraph isomorphism that generalizes subtree isomorphism

## Why this fragment?

1. natural first class beyond trees
  - trees, *outerplanar graphs*, and planar graphs form a natural hierarchy (Hedetniemi, Chartrand & Geller, '71)
2. practically relevant class
  - NCI dataset: **94.3%** (236180 out of 250251) of the compounds are 11-tenuous outerplanar graphs
3. subgraph isomorphism is often not adequate e.g. in chemoinformatics
4. subgraph isomorphism is intractable for outerplanar graphs

# Outerplanar Graphs

- (Chartrand & Harary, '67)
- graphs which can be embedded in the plane in such a way that
  - no two edges intersect except at a vertex in common
  - all vertices lie on the exterior face

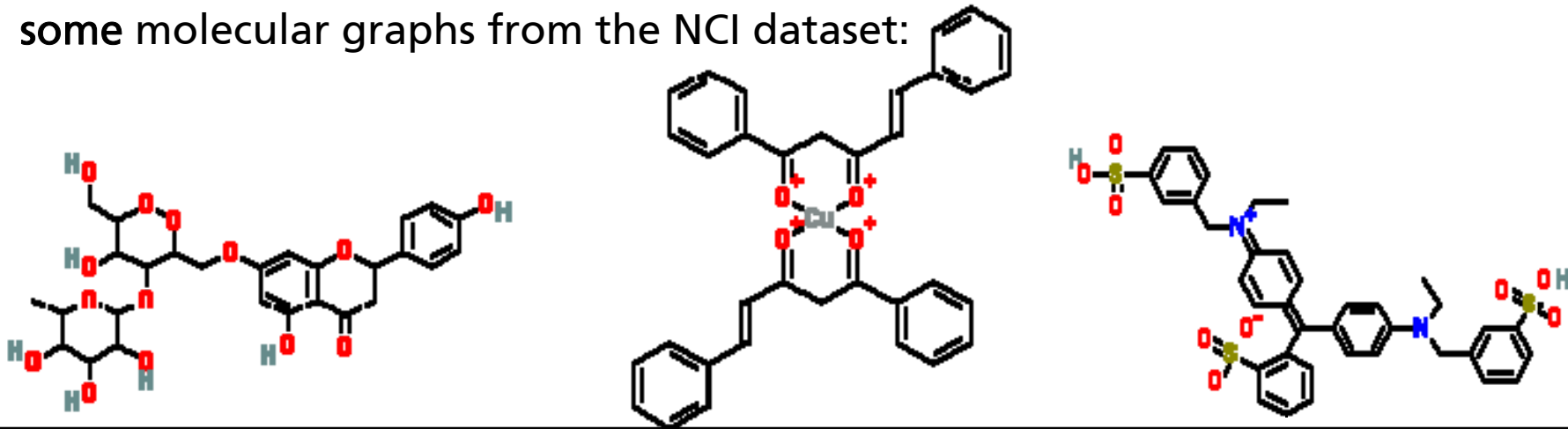


## Properties:

- outerplanarity can be decided in linear time [Mitchell, '79]
- each block (biconnected components) with  $n$  vertices has a unique Hamiltonian cycle
- the unique Hamiltonian cycle
  - can be computed in linear time [Mitchell, '79]
  - has at most  $n-3$  diagonals

# d-Tenuous Outerplanar Graphs

- each block has at most  $d$  diagonals
- NCI dataset:
  - 236180 outerplanar graphs (out of the 250251 compounds)
  - $d = 11$  (only for one compound)
  - $d = 5$  for 236083 (99.99%) outerplanar graphs
- $d$  is considered to be a constant
- some molecular graphs from the NCI dataset:

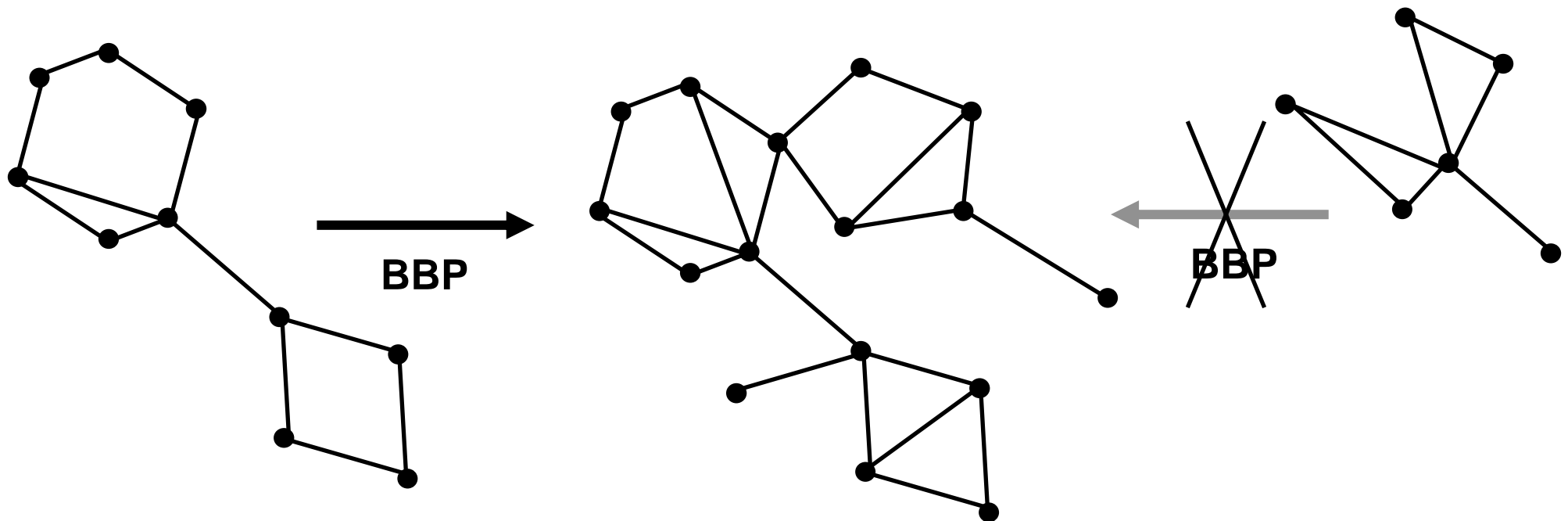


# BBP Subgraph Isomorphism

$G, H$  outerplanar graphs;

a **block and bridge preserving (BBP)** subgraph isomorphism from  $H$  to  $G$  is a subgraph isomorphism from  $H$  to  $G$  mapping

- different blocks of  $H$  to different blocks of  $G$
- bridges of  $H$  to bridges of  $G$



# Mining d-Tenuous Outerplanar Graphs w.r.t. BBP Subgraph Isomorphism

input: set D of d-tenuous outerplanar graphs and  $t > 0$

1. compute the set  $L_1$  of frequent
  - *vertices* and
  - *blocks*
2. compute the set  $L_2$  of frequent
  - *edges* and
  - *pairs of blocks* with a common vertex, and
  - *pairs of block and edge* with a common vertex
3.  $k = 2$
4. **while**  $L_k \neq \emptyset$  **do**
5.    $++k$
6.   generate the set  $C_k$  of candidates from  $L_{k-1}$
7.   compute the set  $L_k$  of frequent patterns from  $C_k$
8. **endwhile**
9. **return**  $\cup_{k>0} L_k$

Precise algorithm

## Main Result

1. canonical string representation for outerplanar graphs ✓
2. computing frequent biconnected graphs ✓
3. candidate generation ✓
4. frequency counting ✓

**Thm:** Frequent  $d$ -tenuous outerplanar graphs can be enumerated in incremental polynomial time.

# Empirical Evaluation - Dataset

NCI dataset [<http://cactus.nci.nih.gov/>]

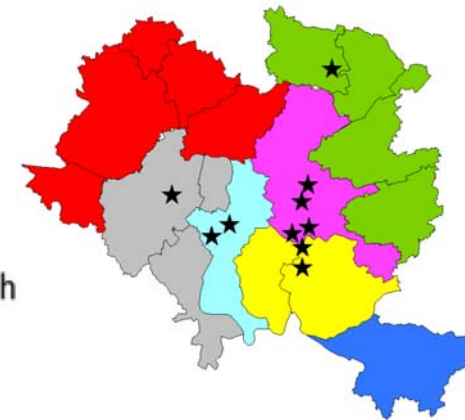
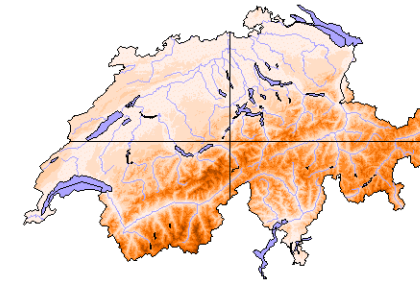
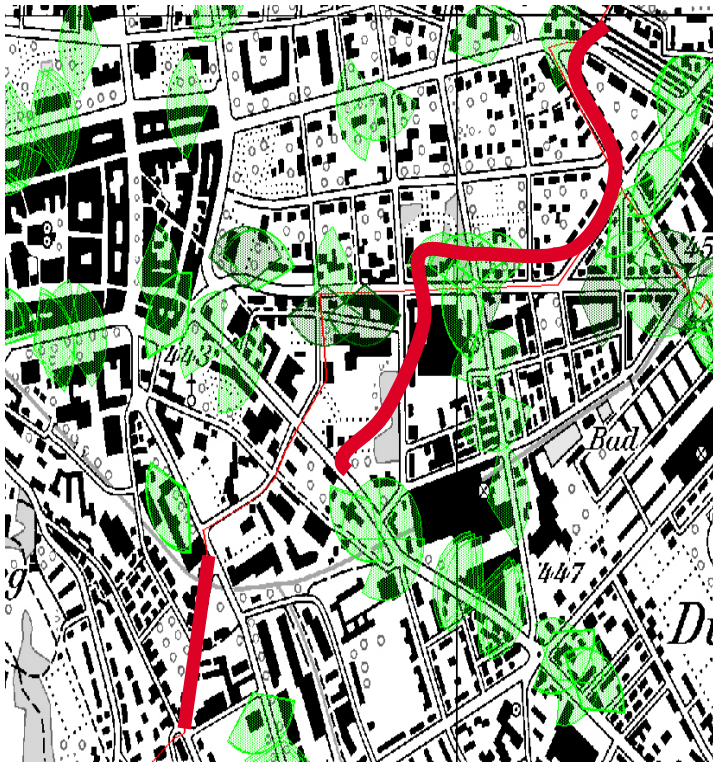
- most frequently used benchmark graph dataset
  - usually small subsets are considered (e.g., HIV)
- 250251 chemical graphs
  - about  $10^7$  compounds have so far been synthesized
- 236180 (i.e., 94.3%) outerplanar
- max number of diagonals ( $d$ ) is small:
  - $d = 11$
  - $d = 5$  for 236083 (i.e., 99.99%)

# Empirical Evaluation - Results

size ( <i>k</i> )	10%			5%			2%			1%		
	#C	#FP	T	#C	#FP	T	#C	#FP	T	#C	#FP	T
1	86	7	107	144	11	169	582	25	380	2196	55	824
2	74	16	446	216	24	570	1332	61	1118	6208	174	2554
3	139	41	1133	234	74	1393	510	170	2123	1516	659	5653
4	133	77	1232	266	154	2038	642	356	4079	2554	1776	11899
5	139	91	1071	319	222	2268	909	644	5603	4550	3886	20411
6	107	72	754	332	252	1847	1212	918	6105	7314	6490	28811
7	61	41	472	295	195	1168	1266	990	4964	10165	9058	34967
8	37	25	354	182	137	741	1086	893	3384	11479	10396	36391
9	20	13	205	137	116	602	956	803	2282	11129	10194	31721
10	8	5	130	131	119	594	828	700	1635	9370	8623	23412
11	0	0	0	131	117	565	697	604	1360	7276	6818	15530
12	0	0	0	115	107	536	707	665	1483	5533	5184	9345
13	0	0	0	78	64	412	1027	1022	2017	4395	4145	5252
14	0	0	0	27	21	250	1702	1700	2858	4303	4194	3707
15	0	0	0	4	3	89	2725	2715	3957	5422	5376	4089
16	0	0	0	0	0	0	4079	4072	5578	7976	7957	5828
17	0	0	0	0	0	0	5518	5487	6898	12742	12729	9077
18	0	0	0	0	0	0	6729	6711	8175	21606	21600	14752
19	0	0	0	0	0	0	7326	7311	8813	37386	37383	24017
20	0	0	0	0	0	0	7114	7079	8703	64241	64238	38821
21	0	0	0	0	0	0	6000	5947	7627	106645	106621	61238
22	0	0	0	0	0	0	4435	4407	5954	168821	168805	93109
23	0	0	0	0	0	0	2857	2855	4129	251406	251278	135048
24	0	0	0	0	0	0	1633	1633	2609	349306	348683	184429
25	0	0	0	0	0	0	787	786	1444	448014	447305	234828

# Efficient Closed Pattern Mining in Strongly Accessible Set Systems

[Boley, Horvath, Poigné, Wrobel/PKDD 2007]



Swiss Poster Research  
s p r

## Example: Track Mining

given a database of GPS-based recordings of spatio-temporal movements (*tracks*), list the set of closed frequent connected subgraphs of movements of people or cars in a street network

- *closed frequent connected subgraphs:*  
    'homogeneous' connected subnetworks

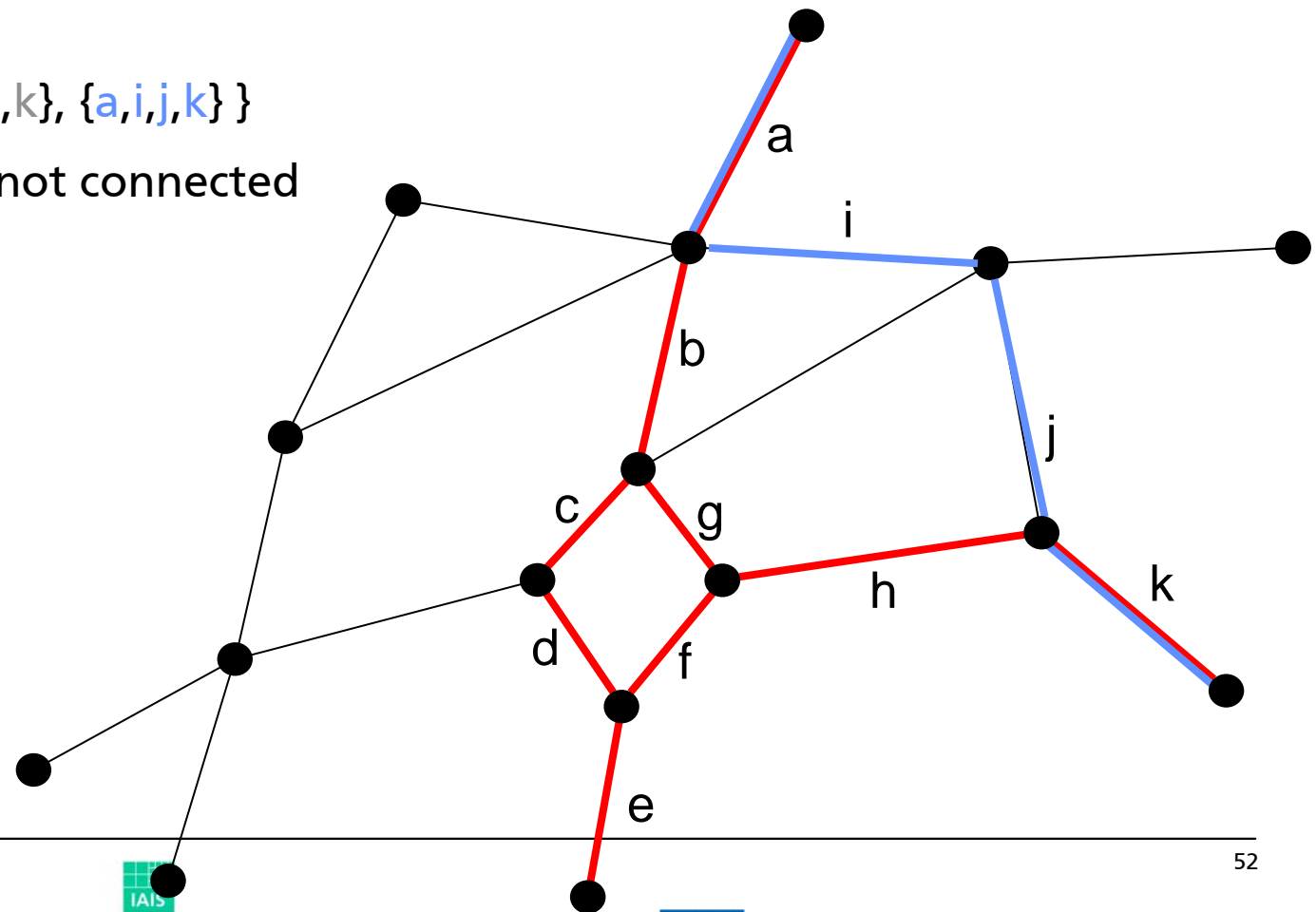
model:

- street network: undirected graph  $G = (V, E)$
- tracks: subsets of  $E$
- embedding operator: subset relation
  - easy to decide
- underlying set system:  $F = \{ X \subseteq E : X \text{ is frequent and connected} \}$ 
  - $F$  is not closed under intersection

# Example

frequency threshold = 1

- $F = \{ \{a,b,c,d,e,f,g,h,k\}, \{a,i,j,k\} \}$ 
  - intersection is not connected

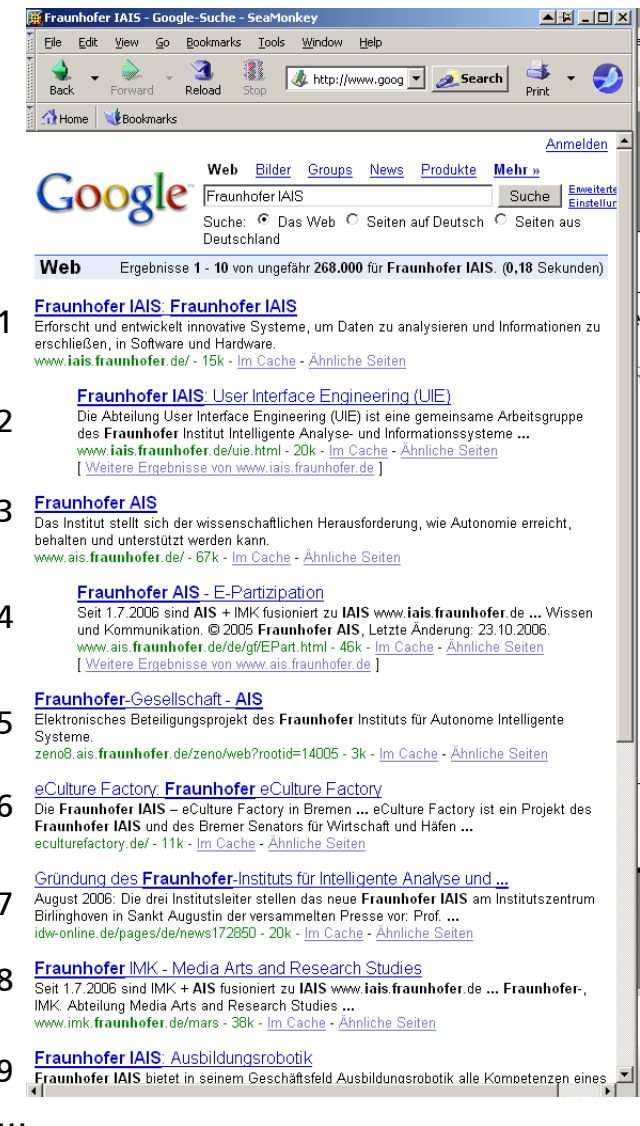


# Outline

- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

# Ranking on undirected graphs

- We are given a set of Web pages
- The user has clicked on pages 3 and 6
  - We interpret this as preferences
    - $p_3 > p_1, p_3 > p_2, p_3 > p_4, p_3 > p_5, p_3 > p_7, p_3 > p_8, p_3 > p_9$
    - $p_6 > p_1, p_6 > p_2, p_6 > p_4, p_6 > p_5, p_6 > p_7, p_6 > p_8, p_6 > p_9$
- in addition, we are given similarities between webpages
  - We interpret this as a similarity graph
    - By using a threshold or weights
- Can we learn the complete preference relation or ranking such that
  - Given user preferences are respected
  - And similar pages are close to each other in the ranking?



# Ranking on graphs - Problem setting

## ■ Input

- Undirected graph  $(V, E), E \subseteq \{\{u, v\} | u, v \in V\}$
- Directed graph  $(V, D), D \subseteq V \times V$

$$\pi : V \rightarrow \llbracket n \rrbracket$$

## ■ Output a permutation

- Few backward edges (directed graph)
- Smooth ordering (undirected graph)

# Ranking on graphs - Optimisation

$$\operatorname{argmin}_{\pi \in V \rightarrow \llbracket n \rrbracket} \sum_{(u,v) \in D} \sigma(\pi(u) - \pi(v)) + \nu \sum_{\{u,v\} \in E} [\pi(u) - \pi(v)]^2$$

$$\text{subject to: } \pi(u) = \pi(v) \Leftrightarrow u = v \quad \forall u, v \in V$$

where  $\sigma$  is the step function and  $\nu$  is a (regularisation) parameter

## Semidefinite ranking on graphs

- Minimum length ordering problem has an SDP-based solution with approximation factor  $O(\log^2|V|)$
- Searches for an embedding of the graph in the Euclidean space of dimension  $|V|$
- Projects the embedding onto a randomly chosen vector
- **Good news:** The geometry of the embedding could easily be exploited to incorporate preference constraints

## The optimisation problem

- Empirical error term

$$\sum_{(u,v) \in D} \langle f(z) - f(u), f(v) - f(u) \rangle + \sum_{(u,v) \in D} \|f(z) - f(u)\| \cdot \|f(v) - f(u)\|$$

- Regularisation term  $\sum_{(u,v) \in E} \|f(u) - f(v)\|^2$

- Summands in the empirical error term are not convex
- First summand becomes convex if we change variables from vectors to inner products between the vectors

# Experiments

- Benchmark metric regression data sets
  - Converted into ordinal regression data sets by discretising the target value into equal-length bins
  - Standardisation (zero mean and unit variance)
  - Similarity graphs using Gaussian kernel
  - Preference constraints encoded in complete bipartite graphs between training instances of successive categories
  - Inverse 5-fold cross validation
- Kendall tau as the evaluation metric
- DSDP for solving SDP-based ranking
- L-BFGS-B for solving spectral ranking

# Results

Dataset	Ins	Pref	Bins	$\tau$ , SDP	Time (in s)	$\tau$ , QP	Time (in s)
Diabetes	43	15	2	$0.60 \pm 0.03$	0.05	$0.58 \pm 0.06$	0.26
Pyrimidines	74	99	5	$0.71 \pm 0.05$	0.50	$0.63 \pm 0.09$	0.26
Triazines	186	284	5	$0.58 \pm 0.04$	3.70	$0.52 \pm 0.01$	0.75
Wisconsin	194	212	5	$0.59 \pm 0.03$	10.02	$0.54 \pm 0.03$	0.06
Machine	209	164	5	$0.76 \pm 0.02$	3.59	$0.73 \pm 0.02$	3.12
Auto	392	1230	5	$0.78 \pm 0.02$	15.59	$0.82 \pm 0.01$	21.48
Housing	506	2043	5	$0.66 \pm 0.02$	67.69	$0.61 \pm 0.02$	23.83
Stocks	950	6736	5	$0.78 \pm 0.02$	204.18	$0.81 \pm 0.04$	187.51

# Outline

- Graphs and Graph Mining
  - graph-structured instances vs. graph-structured spaces
  - Classification, frequent subgraph mining, ranking
- Graph classification using kernels
  - Cyclic pattern kernels [Horvath/Gärtner/Wrobel/04]
- Frequent subgraph mining
  - D-tenuous outerplanar graphs [Horvath/Ramon/Wrobel/06]
  - Closed pattern mining [Boley/Horvath/Poigné/Wrobel/07]
- Ranking
  - Semidefinite ranking [Vembu/Gärtner/Wrobel/07]
- Summary

# Summary

- Graphs and graph mining problems abound
- Most graph problems are proven hard
- Must identify powerful yet tractable classes
  - Outerplanar graphs
  - Strongly accessible set systems
- And adapt algorithms to tasks adding heuristics and approximation
  - Cyclic pattern kernels
  - Frequent subgraph mining
  - Semidefinite ranking

