An Introduction to Graph Mining

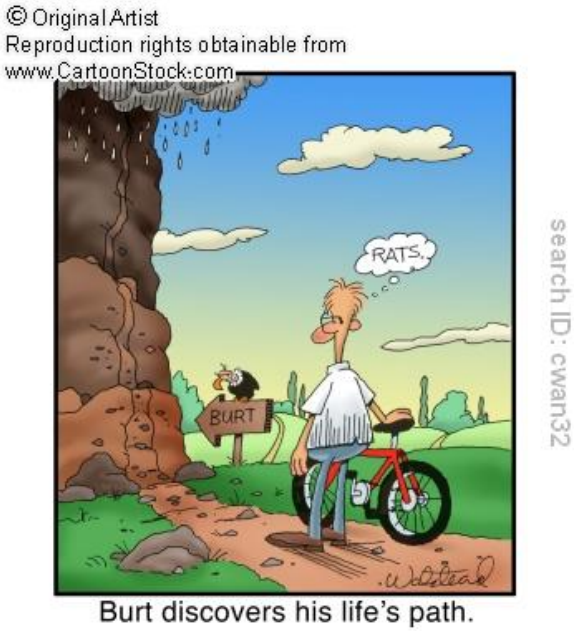
Hossein Rahmani

hrahmani@liacs.nl

8 December 2009

Outline

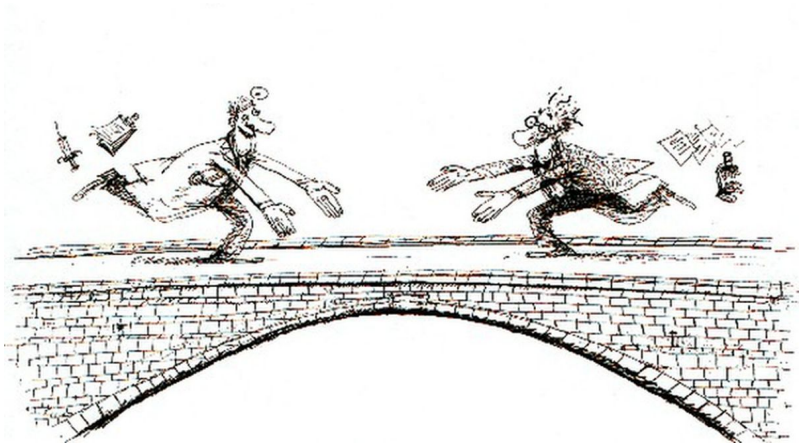
- From Data Mining To Graph Mining
- Graph Algorithms
- Function Prediction in PPI Networks



Burt discovers his life's path.

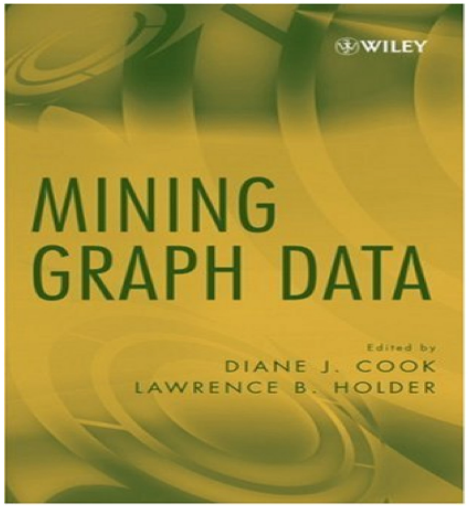
From Data Mining To Graph Mining

- Data Mining
  - Classification
  - Clustering
  - Association rule learning
- Graph Mining
  - Powerful way to represent data
  - output: expressed as graphs



Graph Mining Domains

- Internet Movie Database
- Web Data
- Social Networks Analysis
- Bio-Informatics

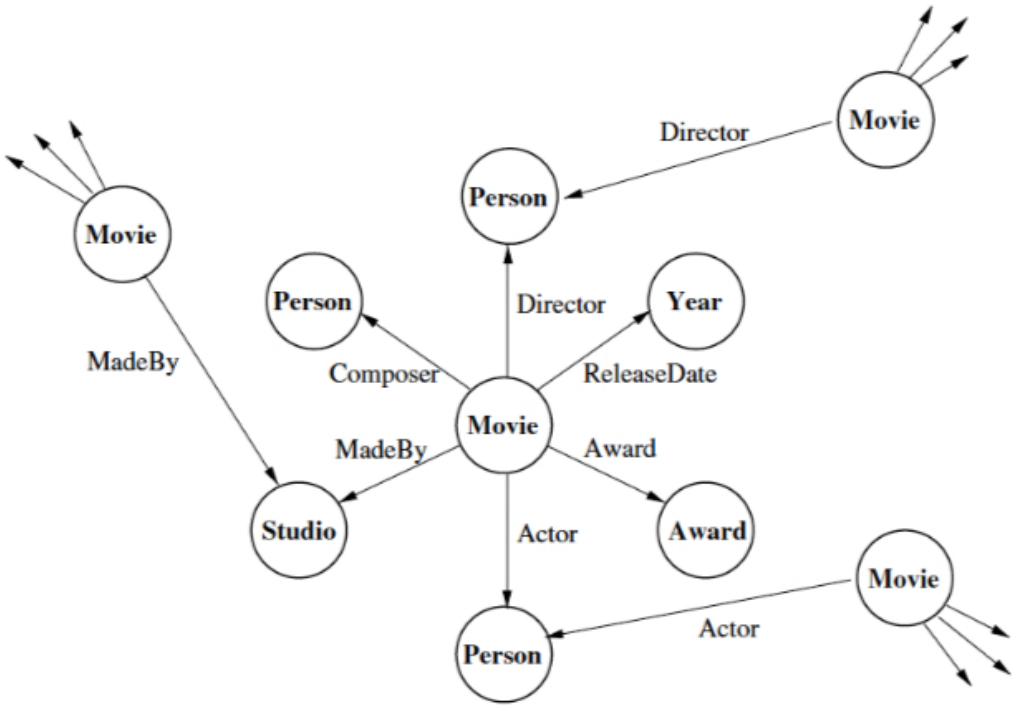


Lawrence B. Holder

Diane J. Cook

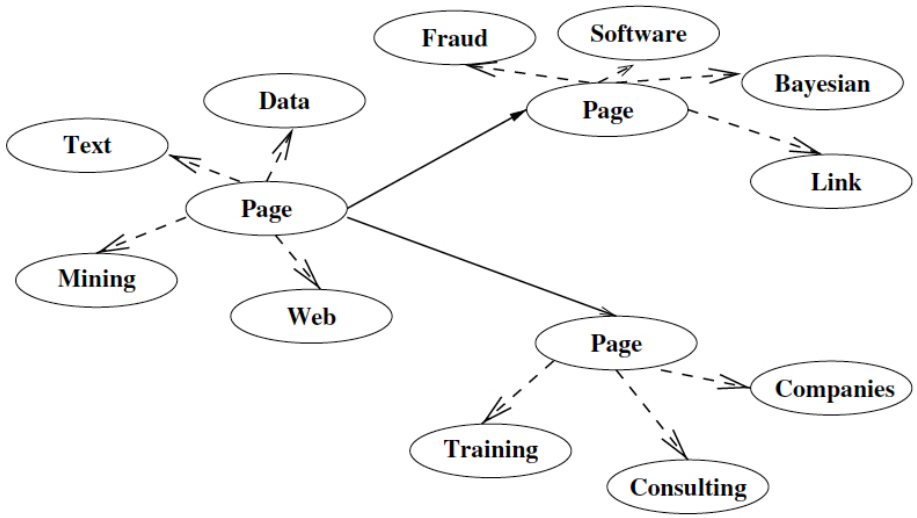
Internet Movie Database

- Movie Recommendation
- Community Detection
- Prediction: $2 million in opening weekend



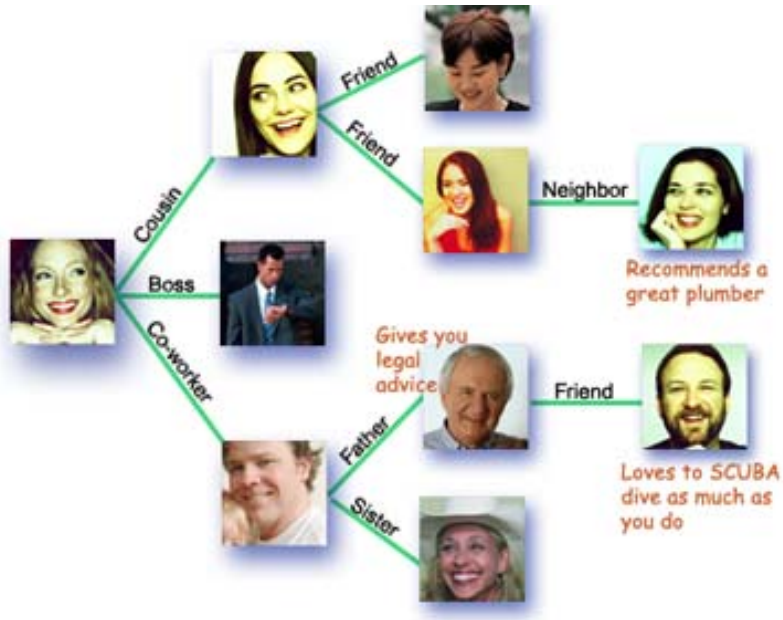
Web Mining

- Web Content Mining
  - Topic Prediction
- Web Structure Mining
  - Community Mining
- Web Usage Mining
  - Website Roadmap



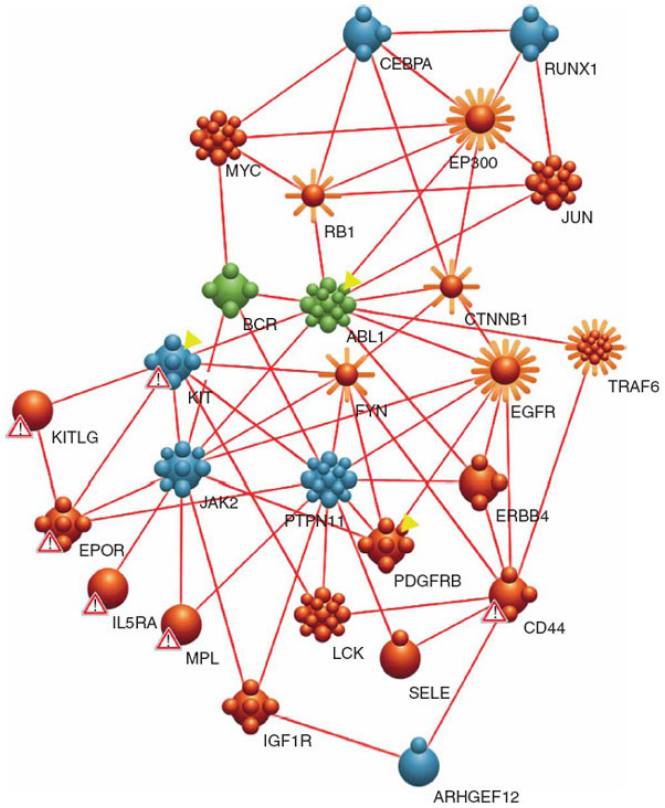
Social Networks

- Relationships and flows between people
- Technologies
  - Email, Blogs
  - Social Networking Software like Orkut, FaceBook
- Questions:
  - Who has control over what flows in the Network?
  - Who has best visibility of what is happening in the Network?
  - Customer Network Value

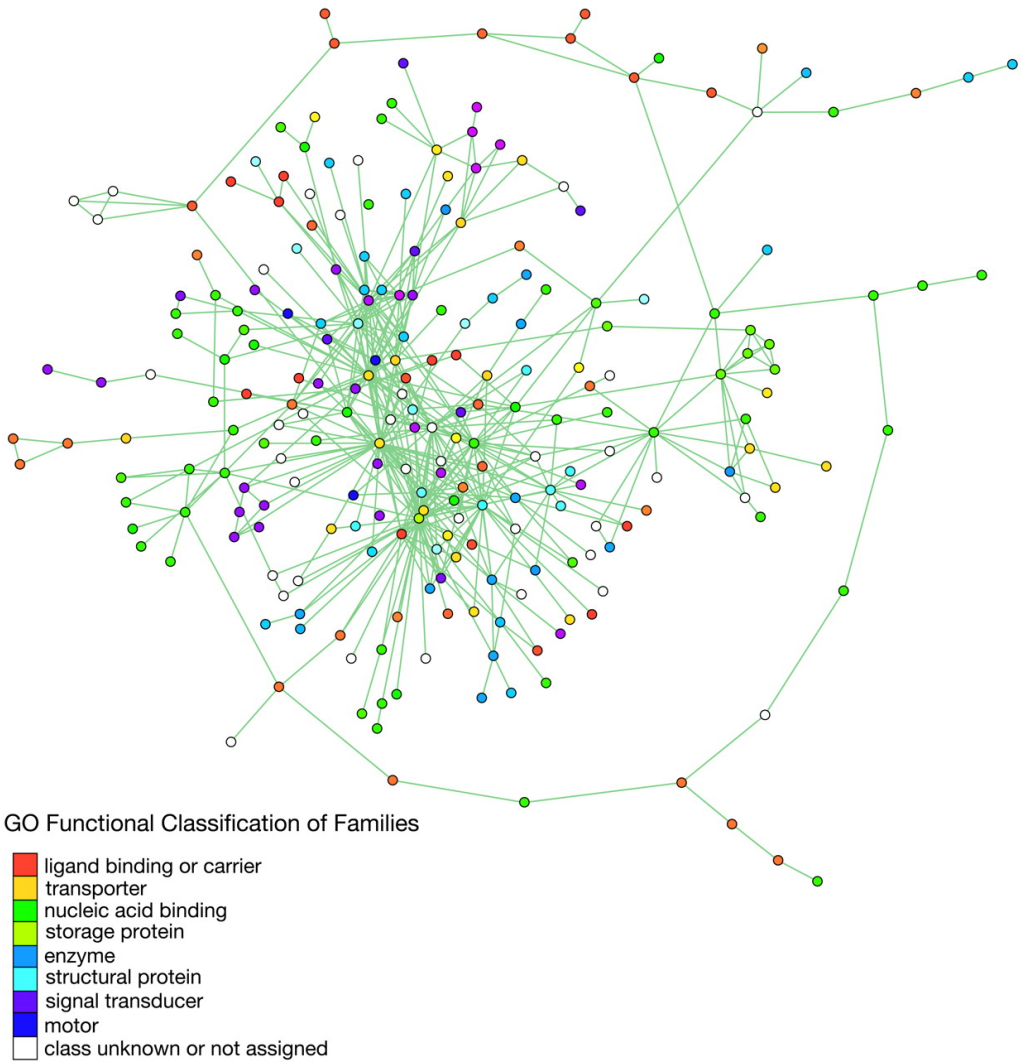


Protein-Protein Interaction(PPI) Network

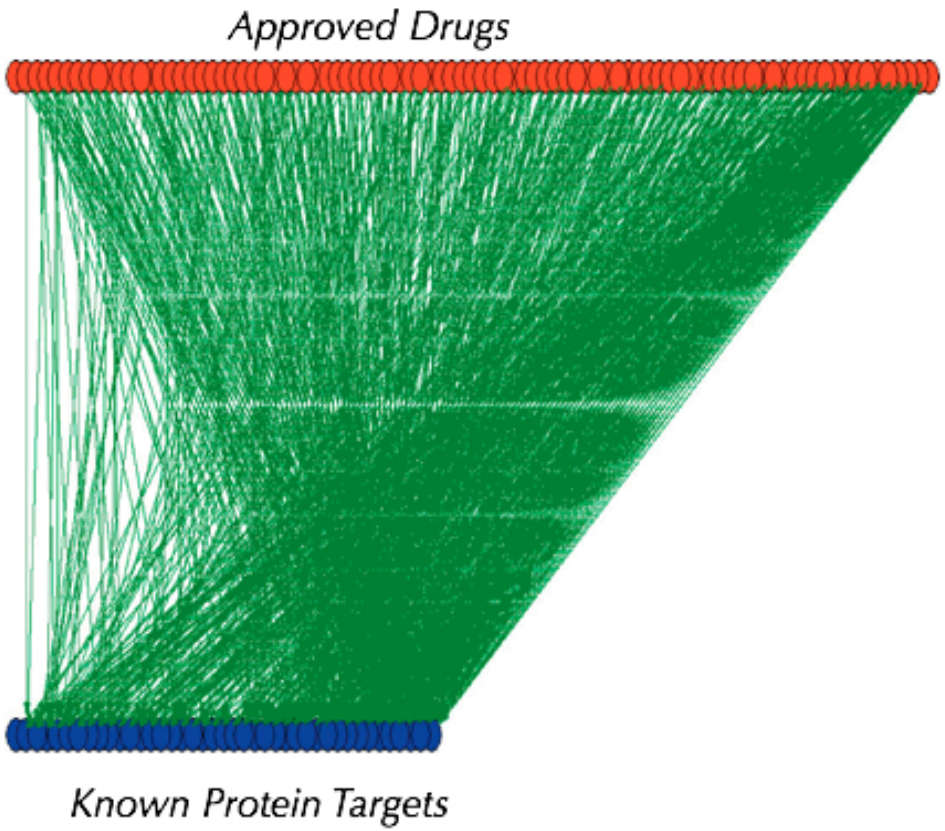
- Nodes: Proteins
- Edges: Interaction among the Proteins
- Open Problems:
  - Function Prediction
  - Drug Discovery



Function Prediction in PPI Networks

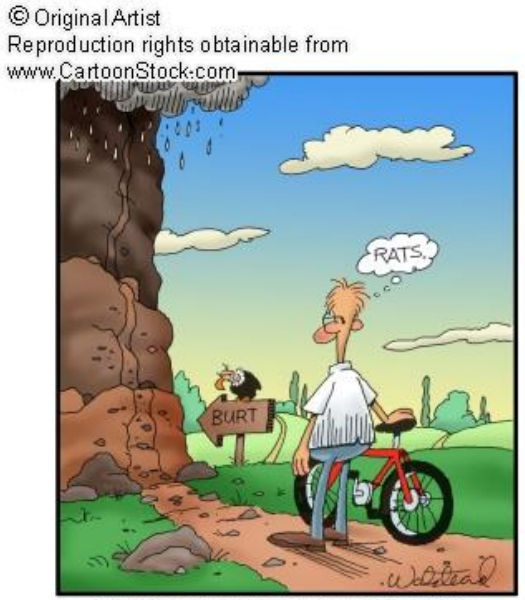


Drug Discovery in PPI Networks



Outline

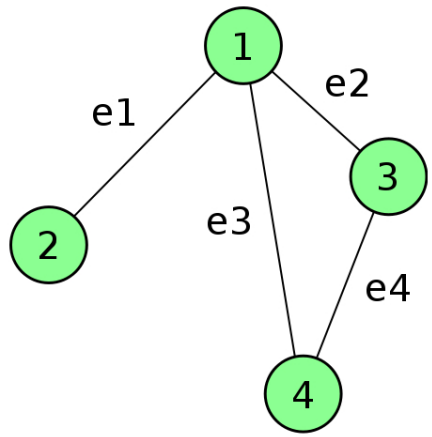
- From Data Mining To Graph Mining
- Graph Algorithms
  - Some Definitions
  - Graph Matching
  - Graph Compression
  - Graph based Decision Tree
- Function Prediction in PPI Networks



Burt discovers his life's path.

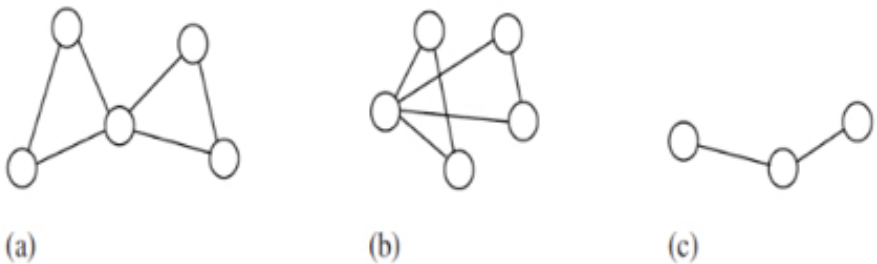
Graph Definition

- Graph: G = (V ,E, μ, v)
  - V : finite set of nodes.
  - $E \subseteq V \times$ V denotes a set of edges.
  - $\mu : V \rightarrow L_V$ denotes a node labeling function.
  - $v : E \rightarrow L_E$ denotes an edge labeling function.
- Let $G_1 = (V_1, E_1, \mu 1, v_1)$ and $G_2 = (V_2, E_2, \mu_2, v_2)$
- Graph G_1 is a subgraph of G_2, written $G_1 \subseteq G_2$, if:
  - $V_1 \subseteq V_2$
  - $E_1 \subseteq E_2$
  - $\mu_1(u) = \mu_2(u)$ for all u $\in V_1$.
  - $v_1(u, v) = v_2(u, v)$ for all (u, v) $\in E_1$.



Graph Definition

- Let $G_1 = (V_1, E_1, \mu_1, \nu_1)$ and $G_2 = (V_2, E_2, \mu_2, \nu_2)$,
- A graph isomorphism between G_1 and G_2 is a bijective function $f : V_1 \rightarrow V_2$ satisfying
  - $\mu_1(u) = \mu_2(f(u))$ for all nodes $u \in V_1$.
  - For every edge $e_1 = (u, v) \in E_1$, there exists an edge $e_2 = (f(u), f(v)) \in E_2$ such that $\nu_1(e_1) = \nu_2(e_2)$.



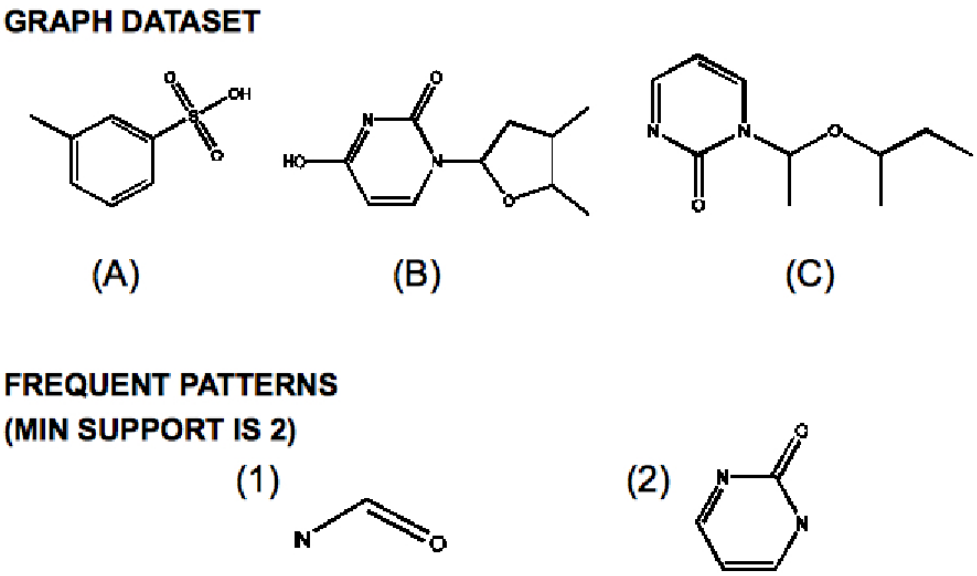
Graph (b) is isomorphic to (a) and (c) is isomorphic to a subgraph of (a)

Graph Definition

- Let G1 = (V_1,E_1,μ_1, v_1) and G_2 = (V_2,E_2,μ_2, v_2)
- Any bijective function $f : \hat{V}_1 \rightarrow \hat{V}_2$, where $\hat{V}_1 \subseteq V_1$ and $\hat{V}_2 \subseteq V_2$ is called edit path from G_1 to G_2.
- Example: $f = \{u_1 \rightarrow v_3, u_2 \rightarrow \epsilon, ..., \epsilon \rightarrow v_6\}$
- $u_1 \rightarrow v_3$: Substitution of node $u_1 \in V_1$ by node $v_3 \in V_2$
- $u_2 \rightarrow \epsilon$: Deletion of node $u_2 \in V_1$
- $\epsilon \rightarrow v_6$: Insertion of $v_6 \in V_2$

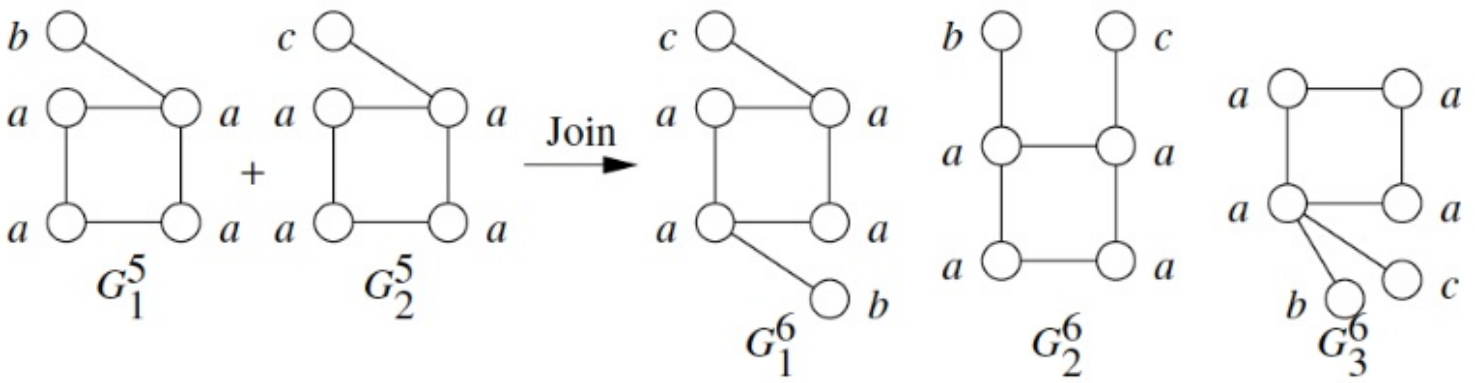
Frequent Subgraph

- Frequent subgraphs:
  - support (subgraph) >= minimum support
- Usage:
  - Graph Classification
  - Graph Clustering
  - Graph Indexing
- Detection Algorithms:
  - Apriori-Based Approach
  - Pattern Growth Approach



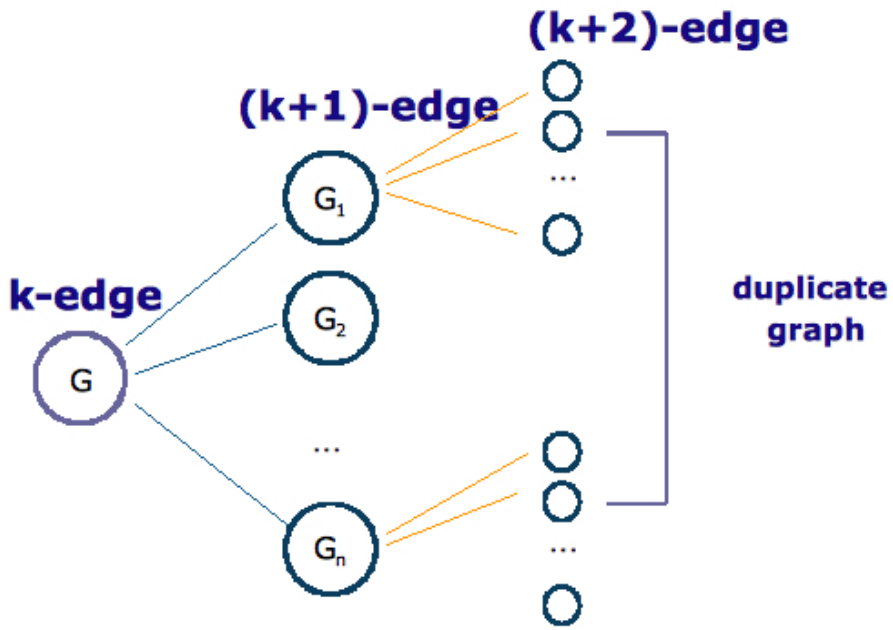
Apriori-Based Approach

```
1: S_{k+1} ← ∅;
2: for each frequent g_i ∈ S_k do
3:   for each frequent g_j ∈ S_k do
4:     for each size (k+1) graph g formed by the merge of
       g_i and g_j do
5:       if g is frequent in D and g ∉ S_{k+1} then
6:         insert g to S_{k+1};
7: if s_{k+1} ≠ ∅ then
8:   call Apriori(D, min_support, S_{k+1});
9: return;
```

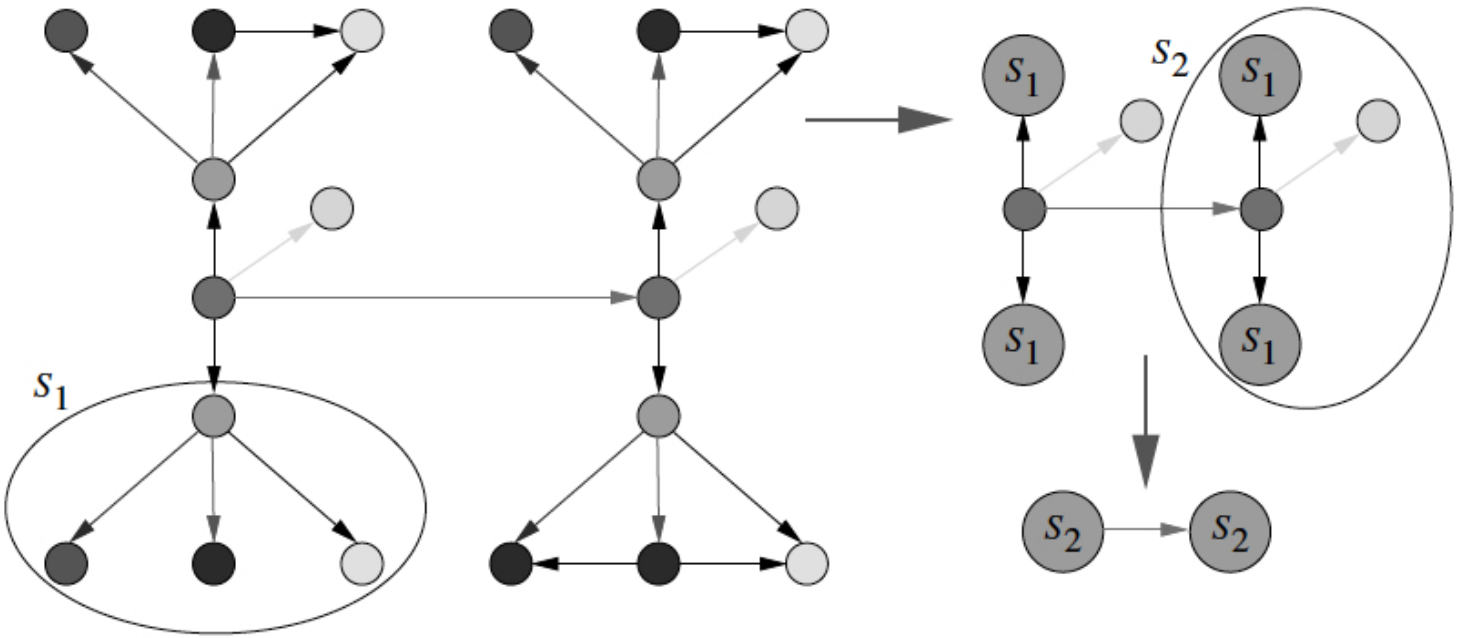


Pattern Growth Approach

- A graph *G* is extended by adding new edge *e*.
- Edge *e* may or may not introduce a new node to *G*.

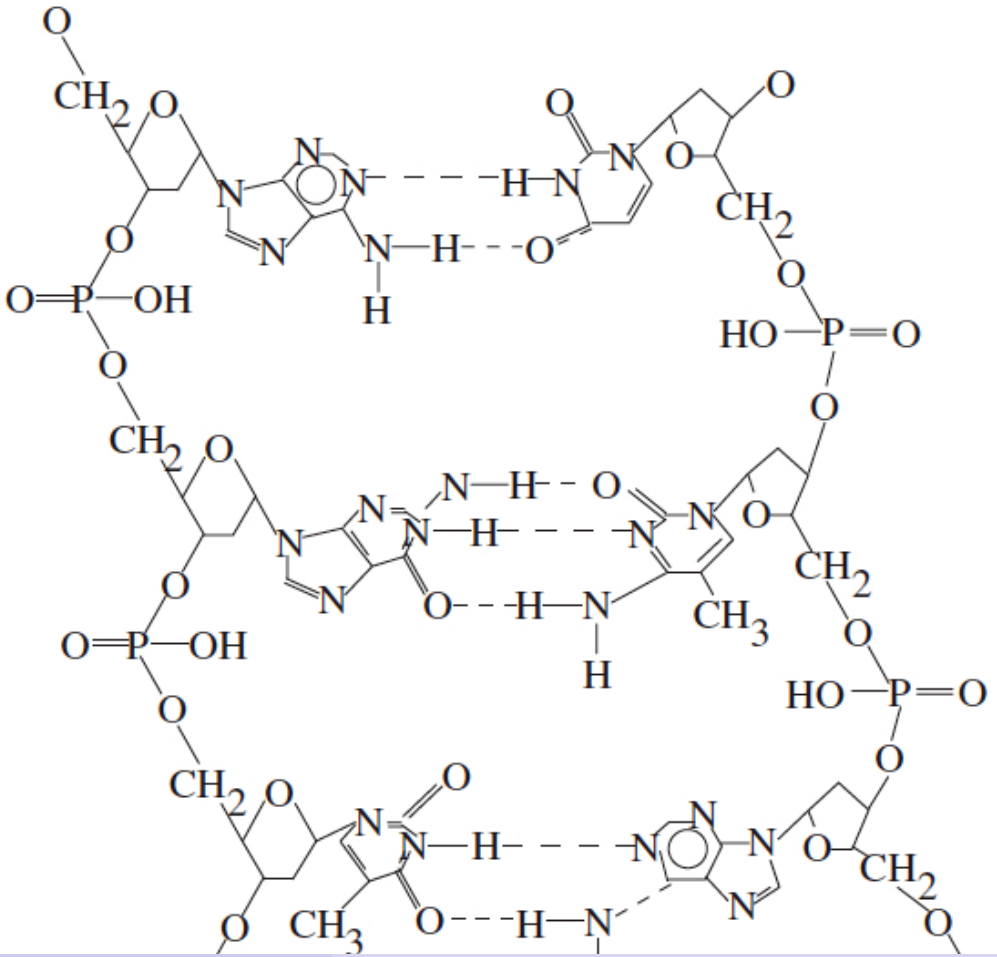


Graph Compression



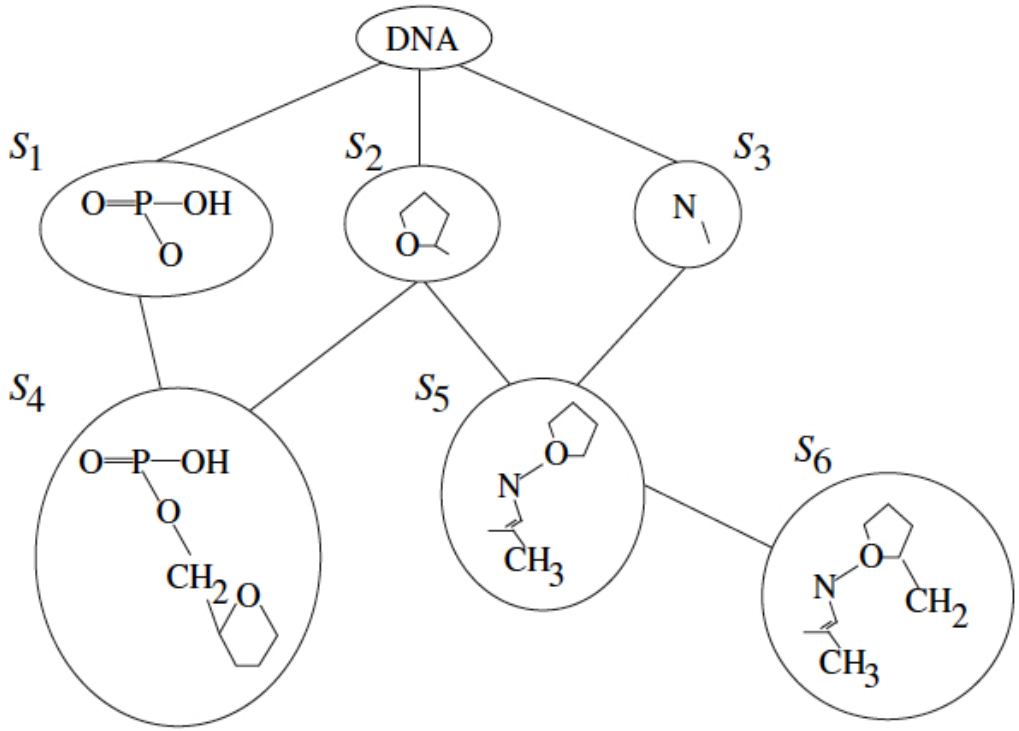
Graph Compression

- Portion of DNA
- Easy to Analyze?



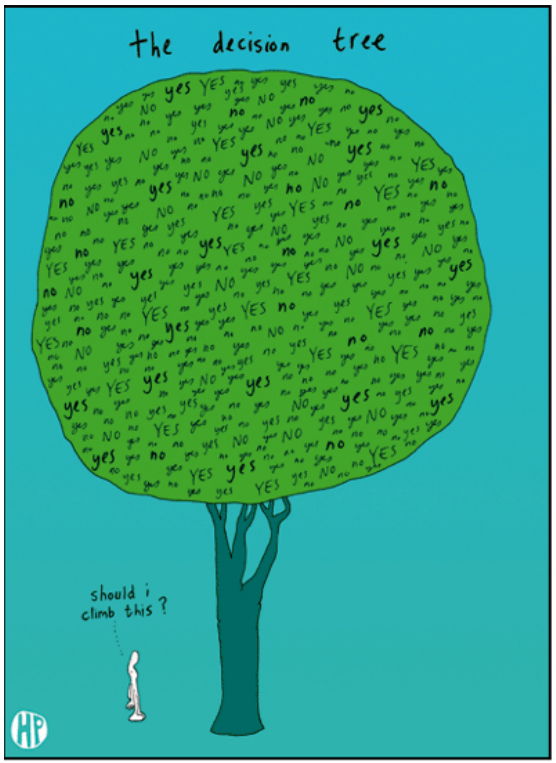
Graph Compression

- S_i compresses the input DNA.
- Cluster hierarchy



Graph Based Decision Tree

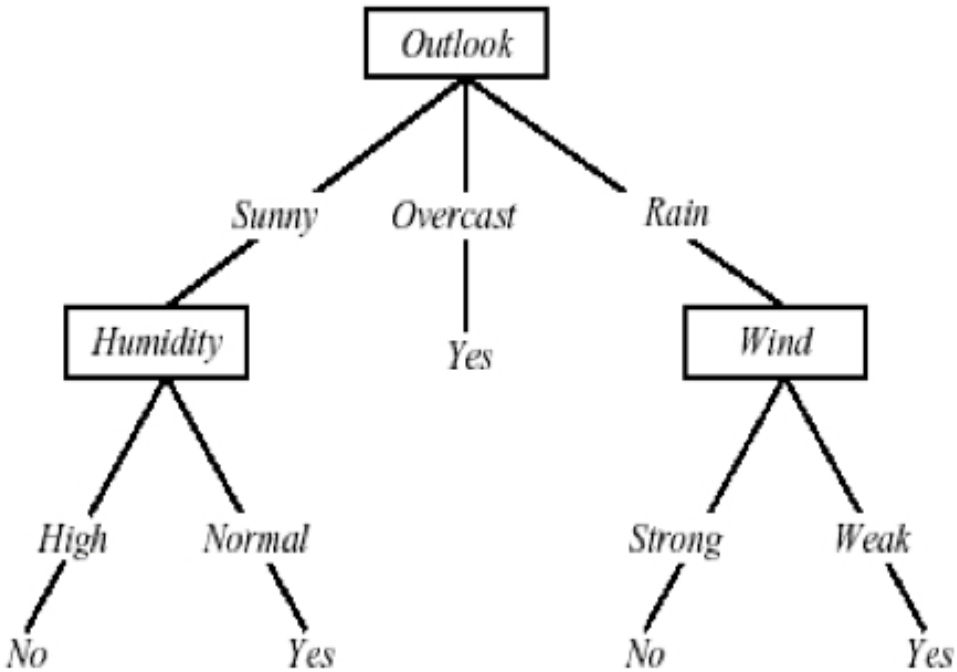
- Decision Tree
  - Each branch corresponds to attribute value
  - Each leaf node assigns a classification



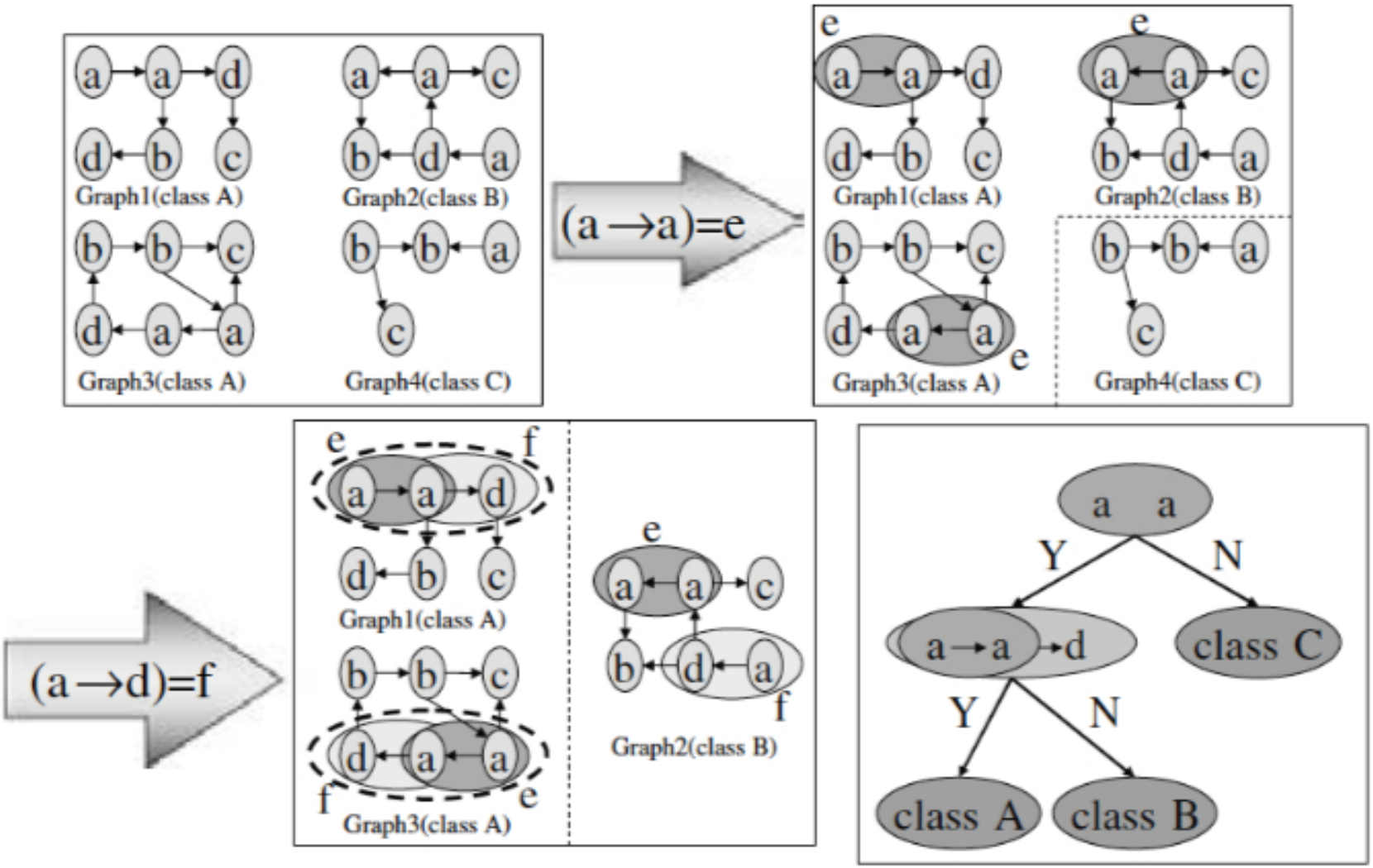
Graph Based Decision Tree

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Graph Based Decision Tree



Graph Based Decision Tree



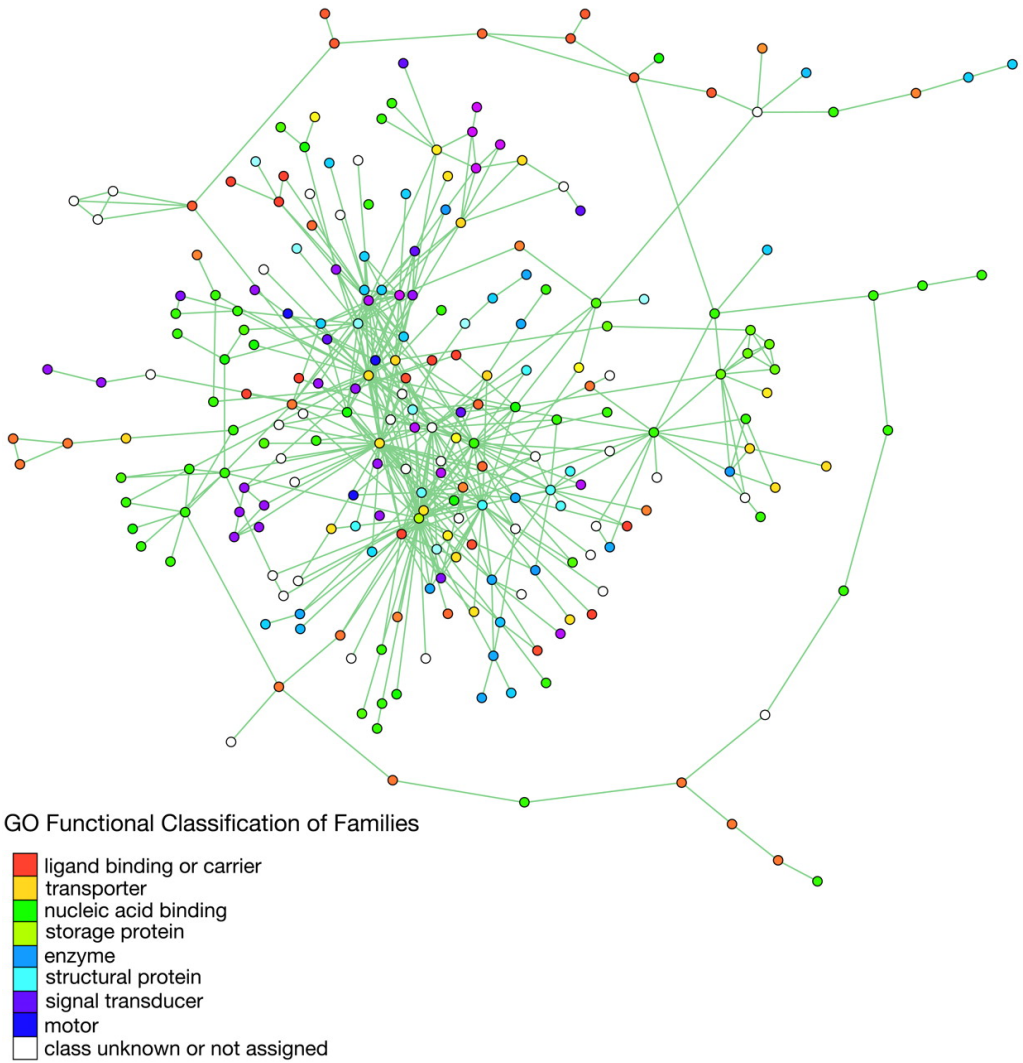
Outline

- From Data Mining To Graph Mining
- Graph Algorithms
- Function Prediction in PPI Networks

Burt discovers his life's path.

search ID: cwan32

Function Prediction in PPI Networks



Formal Description of PPI Networks

- Represented as an undirected graph
  - Node set $V \rightarrow$ Proteins
  - Edge set $E \rightarrow$ Direct interaction
- $\forall v \in V$ is described by a description $d(v) \in D$
  - $d(v)$ derived from the network structure
  - No additional information, such as the protein structure is available
- $\forall v \in V$ optionally is annotated with a label $l(v) \in L$
  - Labels $l(v)$ are sets of protein functions
  - E.g., metabolism, transcription, protein synthesis and etc
- We assume there is a true labeling function λ that is $l(v) = \lambda(v)$ where $l(v)$ is defined
- Task: Find a suitable $\lambda(v)$ where $l(v)$ is not defined

Related Works

1. Transductive approaches
   - Local: Majority Rule and its extensions
   - Global: Global Optimization and Functional Clustering
2. Inductive approaches
   - Local: Topological Redundancies
   - Global: ? $\rightarrow$ Our Method

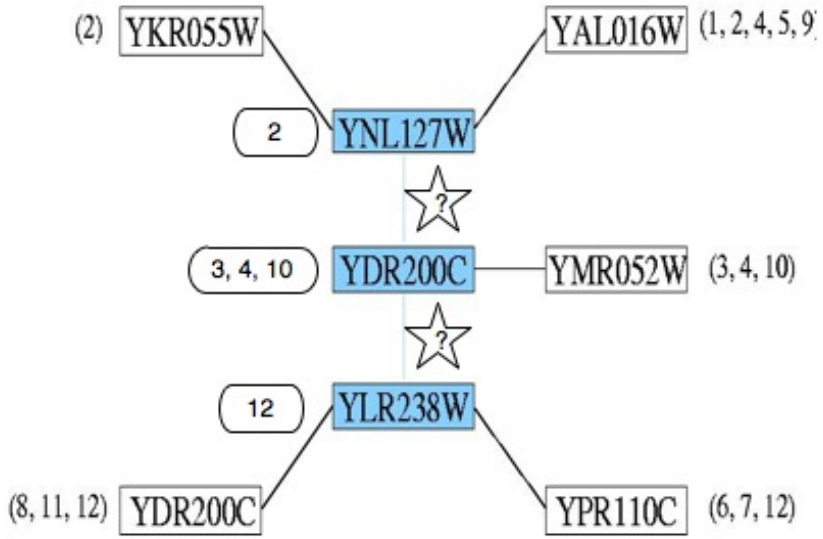
Schwikowski,B

przulj,N

Majority Rule

- Local transductive method
- Assumption: Two Interacting proteins have something in common (e.g., same function)
- Predicted function: Most common function(s) among classified partners

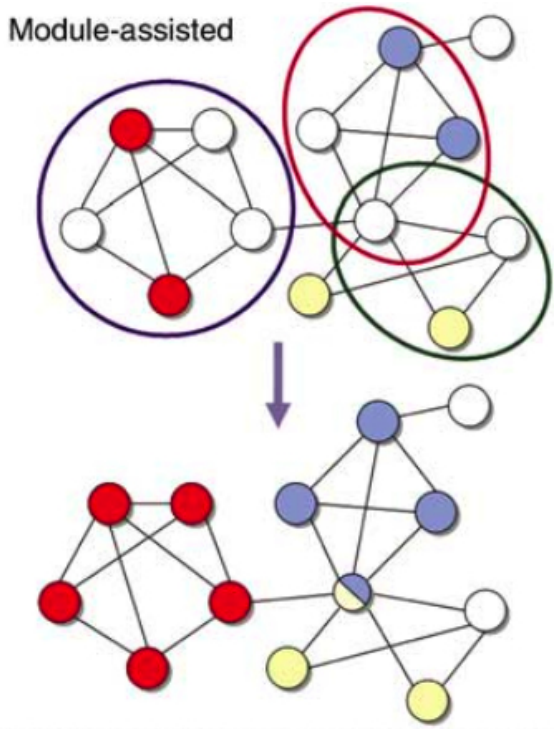
Majority Rule



- Problem: Links unclassified-unclassified proteins completely neglected
- Solution: Global optimization methods

Functional Clustering

- Global transductive method
- Assumption: Dense regions are a sign of the common involvement in biological process
- Predict the function of unclassified protein based on the cluster they belong to



Our Method: A Global Description of Proteins

- Global inductive approach
- Node description
  - N nodes in the network numbered from 1 to N
  - Each node is described by an n-dimensional vector
  - i'th component in the vector of node v gives the length of shortest path between v and node i
  - Probelm: Large Graph $\rightarrow$ very high dimensional descriptions
  - Solution: Reduce dimensionality by focusing on shortest-path distance to a few "important" nodes
    - Feature selection problem

Rahmani, H

Blockeel, H

Bender, A

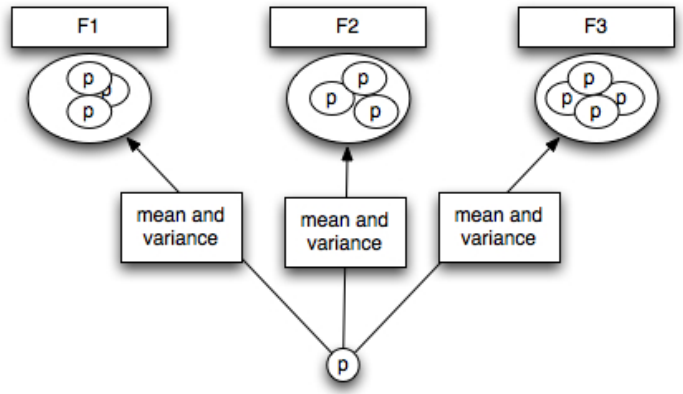
Important Proteins

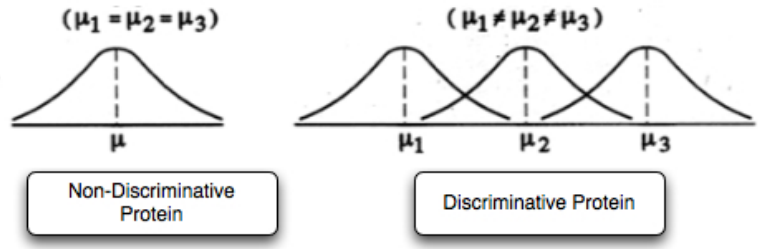
- Definition: Node i is important if the shortest-path distance of some node v to node i is likely to be relevant for v's classification
- Feature f_i denotes the shortest-path distance to node i
- Anova based feature selection
  - For each function j, let G_j be the set of all proteins that have that function j
  - Let $\bar{f}_{ij}$ be the average f_i value in G_j
  - Let $var(f_{ij})$ the variance of the f_i in G_j
  - Anova (analysis of variance) based relevancy measure:

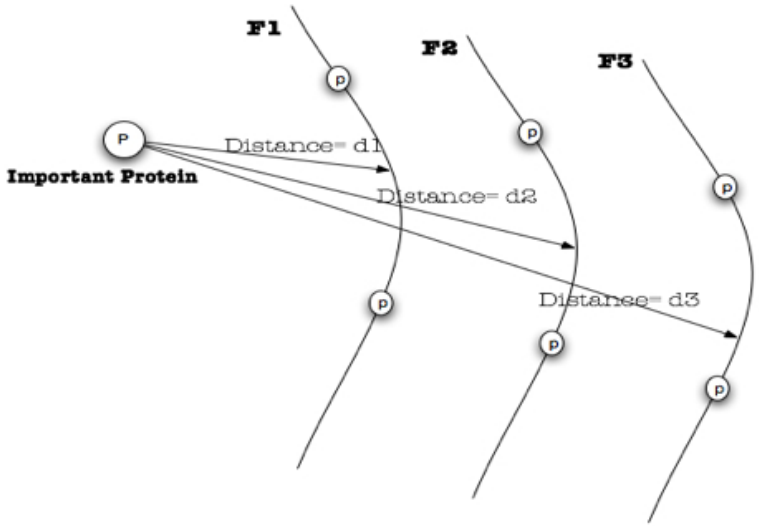
$$A_i = \frac{Var_j[\bar{f}_{ij}]}{Mean_j[var(f_{ij})]} \quad (1)$$

  - A high A_i denotes a high relevance of feature f_i

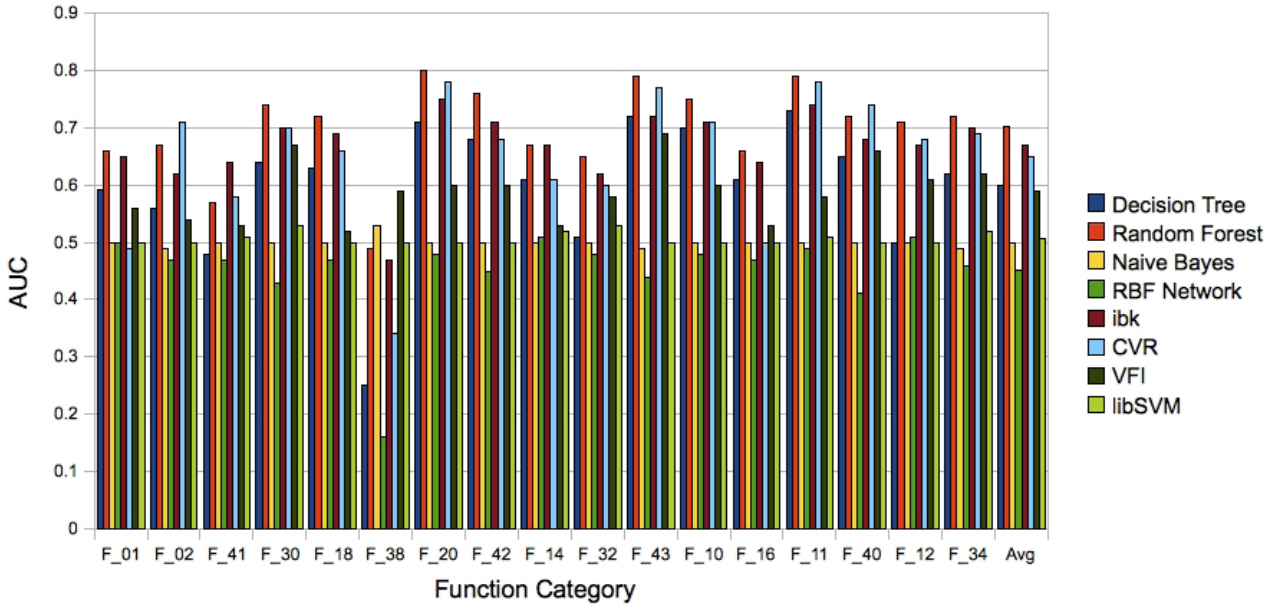
Important Proteins







Comparison of Learners



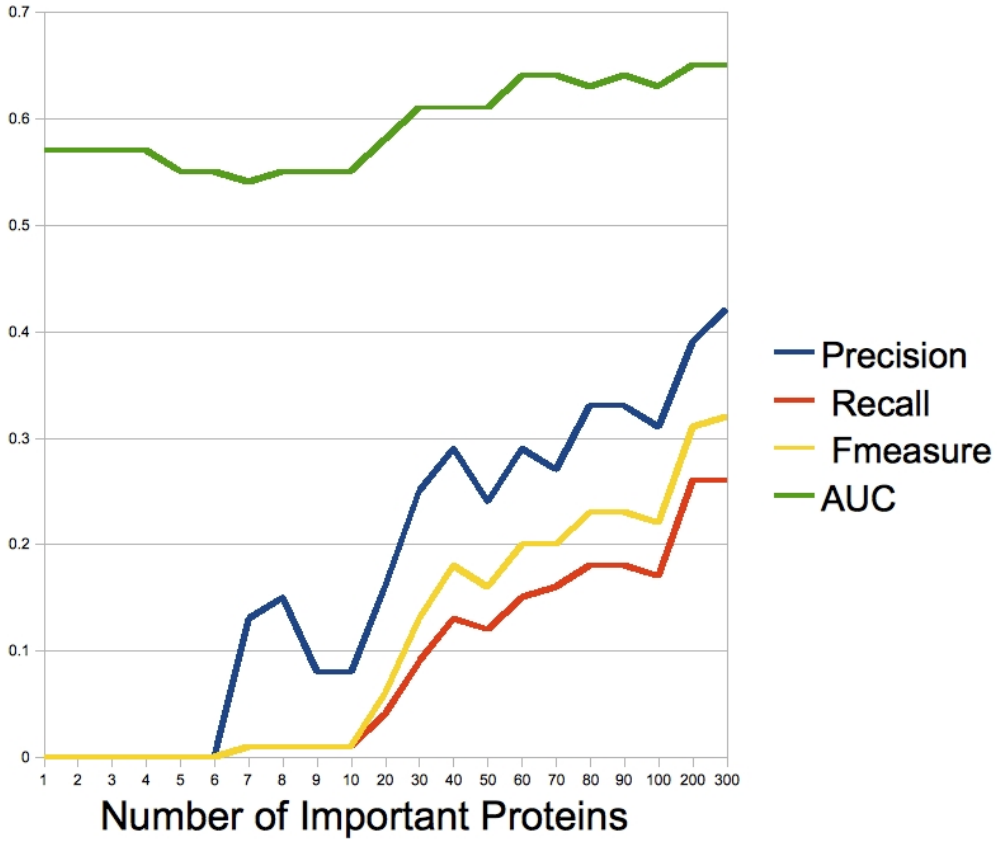
- Random Forest performs best among all the learners in 13 out of 17 cases
- Other 4 cases are all characterized by a very high class skew
- Random Forest: Best candidate for learning from this type of data

Different Number of Important Proteins

- We select the 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200 and 300 most important proteins.
- Describe each protein using its distance to the n most important proteins.
- Using random forest for function prediction and record precision, recall, F-measure and AUC.
- In 4 Datasets for 17 functions.
- The shape of the curves is qualitatively very similar in all cases.

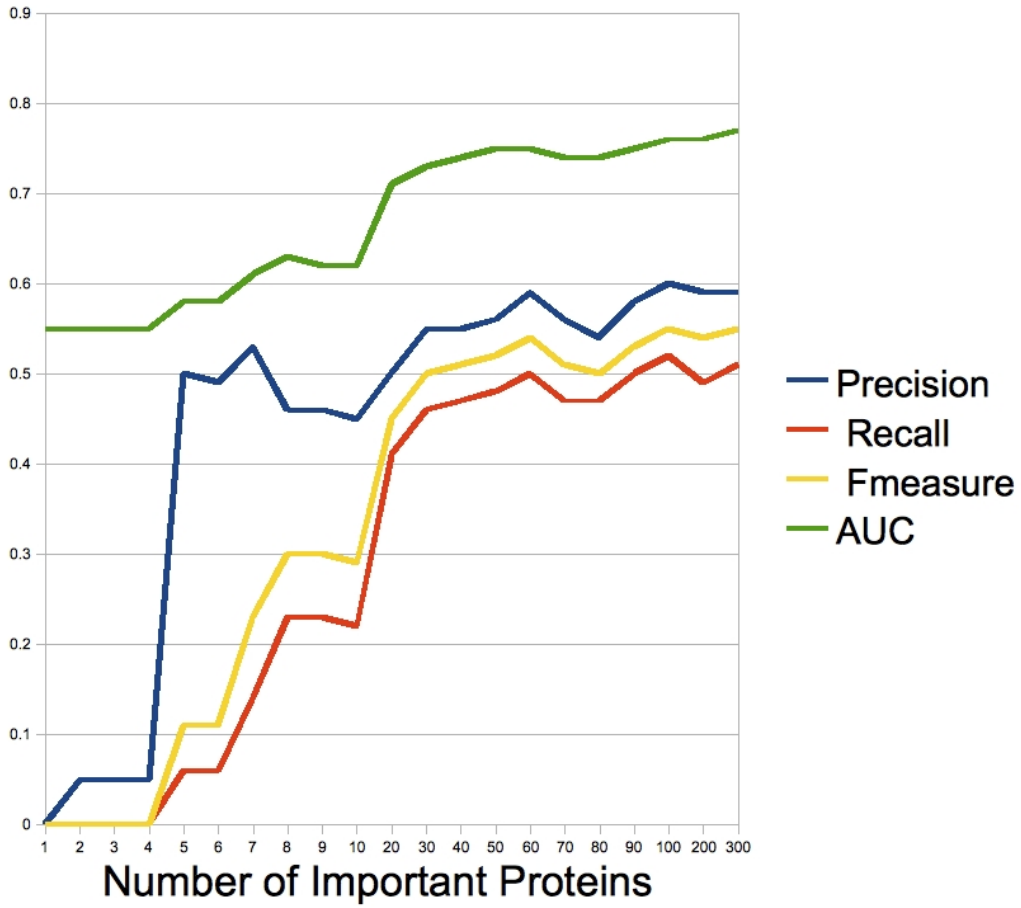
Less than 10 Important Proteins

- Bad Bad predictive performance.
- They do not reach their maximum.



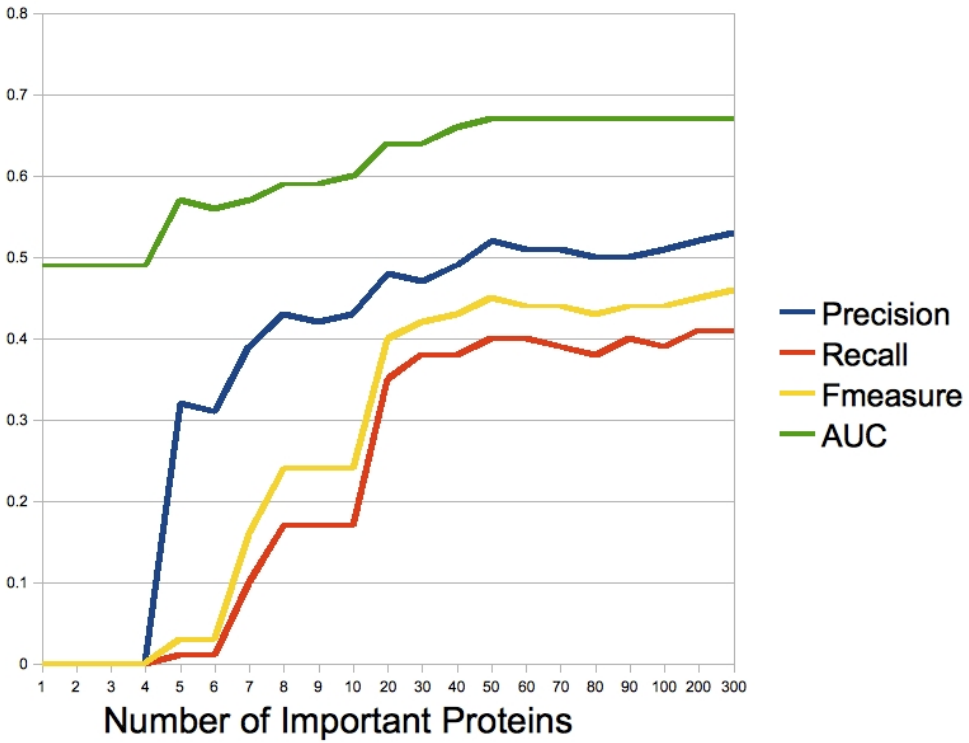
10-50 Important Proteins

- There is usually a major improvement in all four metrics.



50-70 Important Proteins

- For most of the functions, selecting 50-70 important proteins is enough to obtain good classication results.



Conclusions

- Graph Mining Domains
- Introduction to Graph Algorithms
  - Graph Matching
  - Graph Compression
  - Graph based Decision Tree
- Protein Function prediction

Thanks!

References

- Cook, D. J. and Holder, L. B. 2006 Mining Graph Data. John Wiley & Sons.
- Rahmani, H., Blockeel, H. and Bender, A., Proc. 3rd Int. Workshop in Machine Learn. Syst. Biol. (MLSB09) 2009 85 – 94.

Transductive Learning

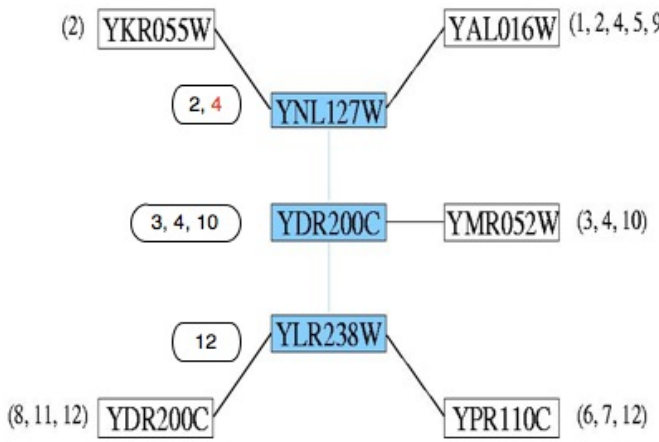
- Task: Predict the label of all the nodes
- Input: $G = (V, E, d, l)$ with l a partial function
- Output: Complete version $G' = (V, E, d, l')$ with l' a complete function that is consistent with l
- l' should approximates λ by optimization criterion o
- o expresses our assumption about λ
  - E.g., directly connected nodes tend to have similar labels
  - Number of $\{v_1, v_2\}$ edges where $l'(v_1) \neq l'(v_2)$ edges should be minimal
- Our assumption about λ is called bias of transductive learner

Inductive Learning

- Task: Learn a function $f : D \rightarrow L$ that maps a node description $d(v)$ onto its label $l(v)$
- Input: $G = (V, E, d, l)$ with l a partial function
- Output: $f : D \rightarrow L$ such that $f(d(v)) = l(v)$
- Note: f differs from l in that it maps D, not V, onto L
  - It can make prediction for node v that is not in the original network, as long as $d(v)$ is known
- Biases
  - Transductive bias: Assumption expressed by optimization criterion o
  - Description bias D
  - Inductive bias: Choice of learning algorithm that is used to learn f from $(d(v), l(v))$ couples

Global Optimization

- Global transductive method
- Links unclassified/unclassified proteins also taken into account
- Any probable function assignment to the whole set of unclassified proteins is considered
  - Counting number of interacting pairs with no common functions
  - Select the function assignment with lowest value

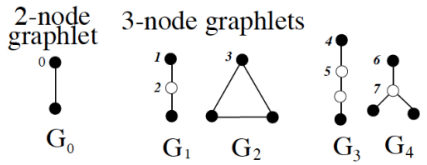


Topological Approaches

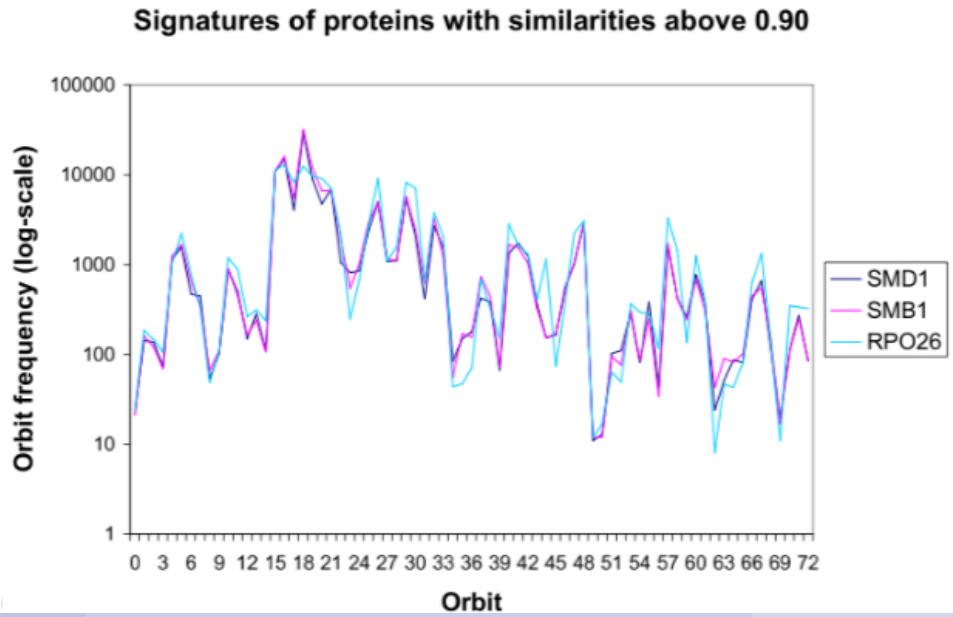
- Local inductive method
- Node description $d(v)$ is built based on the local neighborhood
- Count number of patterns (e.g., graphlet) around the proteins
- Make the signature vector for each protein
- Assumption: Proteins with high similar signature vector have same functions

Topological Approaches

- Some topological patterns (Number of considered patterns = 73)



- Orbit: One of the previous patterns
- Same orbit frequency $\rightarrow$ same function

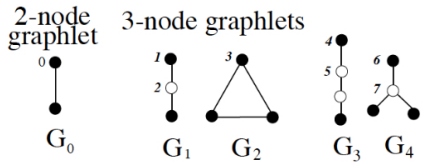


Topological Approaches

- Local inductive method
- Node description $d(v)$ is built based on the local neighborhood
- Count number of patterns (e.g., graphlet) around the proteins
- Make the signature vector for each protein
- Assumption: Proteins with high similar signature vector have same functions

Topological Approaches

- Some topological patterns (Number of considered patterns = 73)



- Orbit: One of the previous patterns
- Same orbit frequency $\rightarrow$ same function