Data-driven Intelligent Systems

Lecture 22
Mining Structure from Graphs II

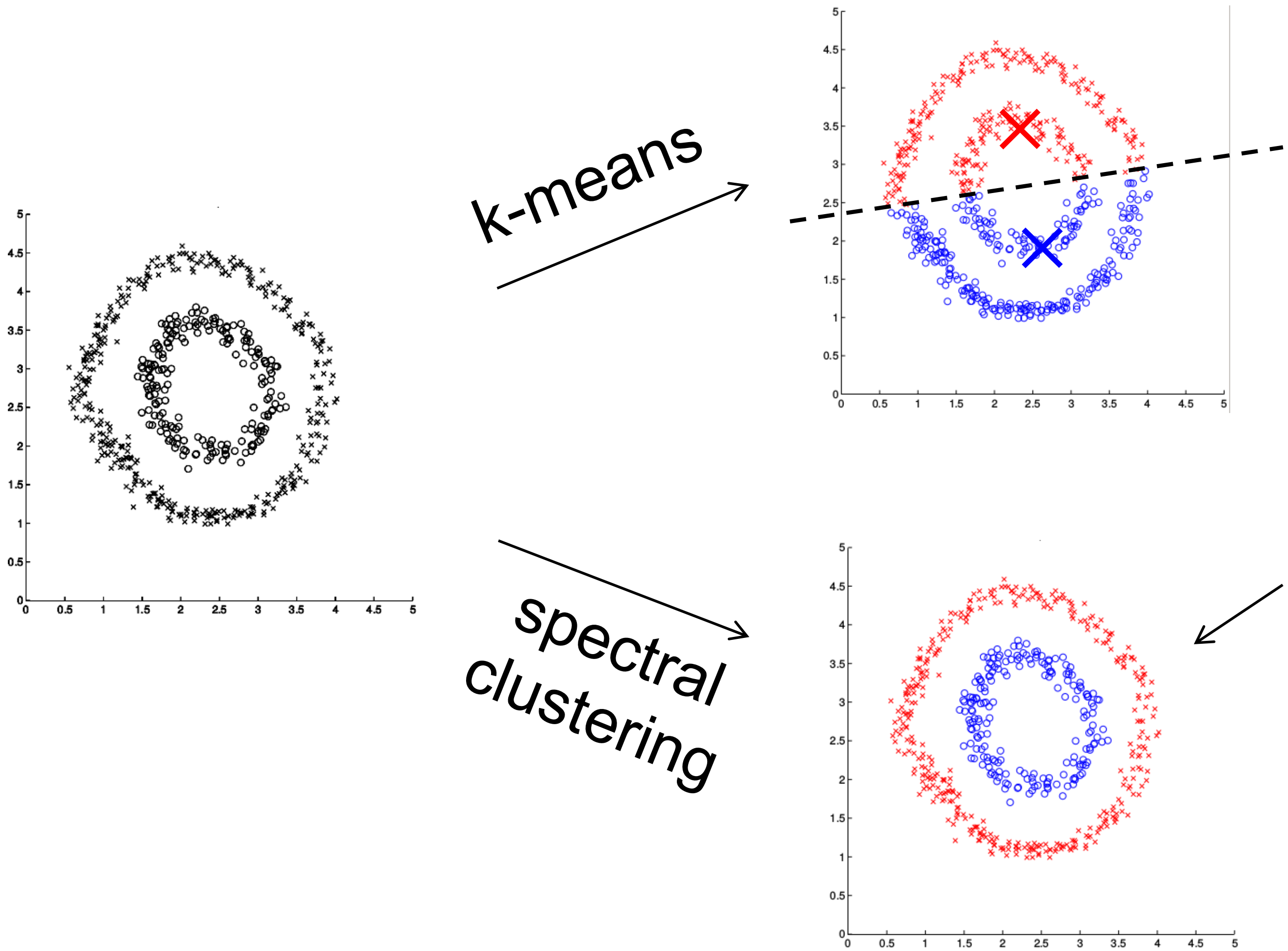
KNOWLEDGE
TECHNOLOGY

http://www.informatik.uni-hamburg.de/WTM/

Overview

- Spectral Clustering
- Semantic Networks
- Bayesian Belief Networks

Spectral Clustering

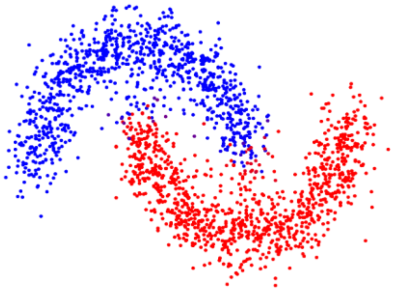


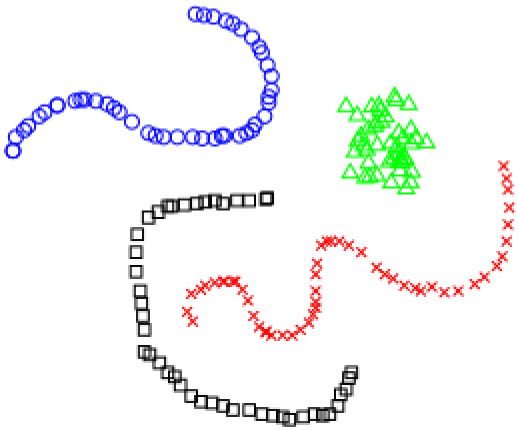
here, 2 centers
and a linear
boundary

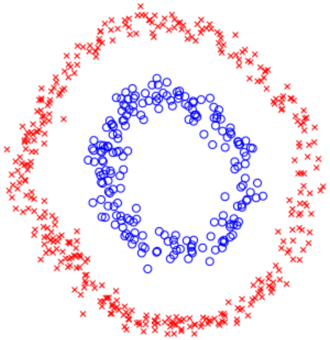
would not be
possible with 2
centers and
linear boundary

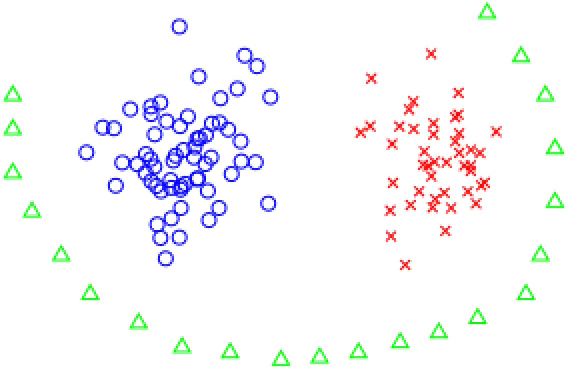
Ng, Jordan, Weiss. On Spectral Clustering:
Analysis and an algorithm. NIPS, 2001

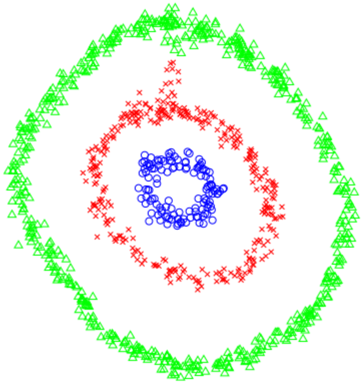
Spectral Clustering





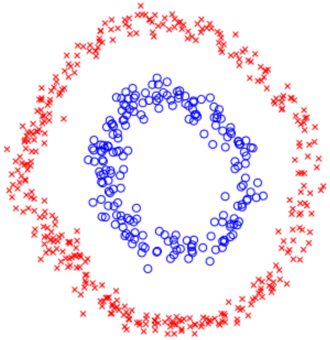






Ng, Jordan, Weiss. On Spectral Clustering: Analysis and an algorithm. NIPS, 2001

Spectral Clustering Idea I/II



- Information about the distances between any two data points will be needed

  $\rightarrow$ similarity (affinity) matrix $A \in R^{nxn}$

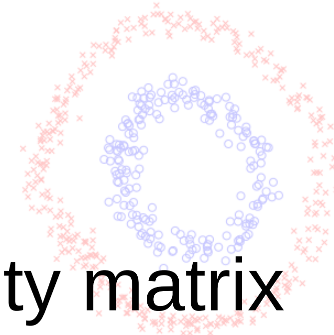
  where n = number of data points

  - Affinities A_{ij} large, if data points i and j nearby
  - $A_{ij} \approx 0$, if i and j far apart (in different clusters)
  - $A_{ij} = A_{ji}$ i.e. A is symmetric

    $\rightarrow$ Eigenvectors orthogonal

    (for unequal eigenvalues)

Spectral Clustering Idea II/II



- Uses spectrum (all eigenvalues) of affinity matrix
  $A \in R^{nxn}$ (*n* = number of data points)
- Mapping to R^k reduces dimensionality

  (*k = number of clusters; k << n*)
- It will turn out: mapped to eigenvectors in R^k, data will form tight clusters at 90° w.r.t each other
- Ng, Jordan, Weiss (2001) model
  one of many variations

Ng, Jordan, Weiss. On Spectral Clustering: Analysis and an algorithm. NIPS, 2001

Spectral Clustering Algorithm

- Given a set of points $S = \{s_1, \ldots, s_n\} \in R^l$
- Aim: compute direct *k*–way partitioning
  - (NOT: generate two clusters, then recurse to generate more)
  - Most parts of the spectral clustering algorithm do not need *k* to be specified (just: $k \leq$ n)

Spectral Clustering Algorithm

- Form the affinity matrix $A \in R^{nxn}$
- Define $A_{ij} = e^{-\|s_i - s_j\|^2 / 2\sigma^2}$ if $i \neq j$ else $A_{ii} = 0$
  - Scaling parameter σ chosen by user
  - Actual data values get lost – only proximities count!
- *Only for normalisation,* define *D* a diagonal matrix whose *(i,i)* element is the sum of *A*'s row *i*
  - D_{ii} is large for elements *i* that have many large A_{ij} (i.e. many neighbours)

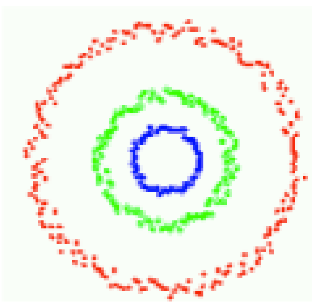
Spectral Clustering Algorithm

- Normalized affinity matrix: $L = D^{-1/2} A D^{-1/2}$
- Find $x_1, x_2, \ldots, x_k$, the *k* eigenvectors of *L* that belong to the *k* largest eigenvalues
  - these eigenvectors show to directions of largest, 2nd-largest, …, k^{th}-largest covariance
- These form the *k* columns of the new matrix *X*

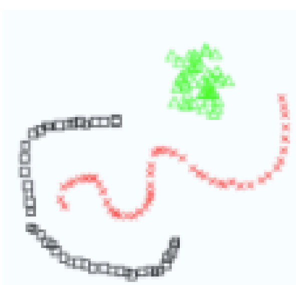
$\longrightarrow$ dimension reduces from $n^x n$ to $n^x k$

Spectral Clustering Algorithm

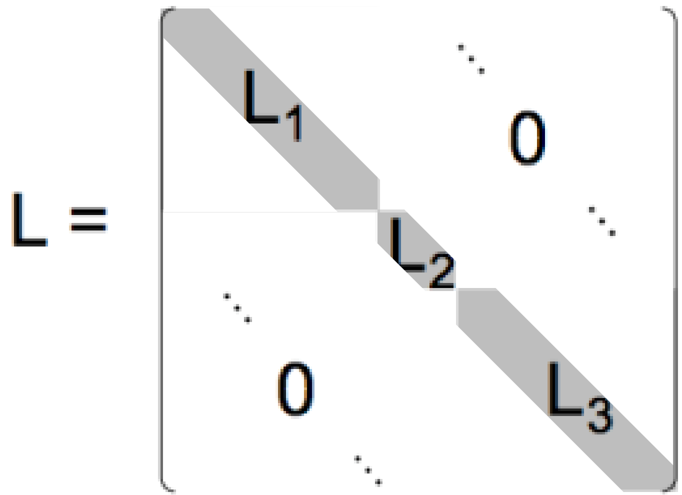
data



OR

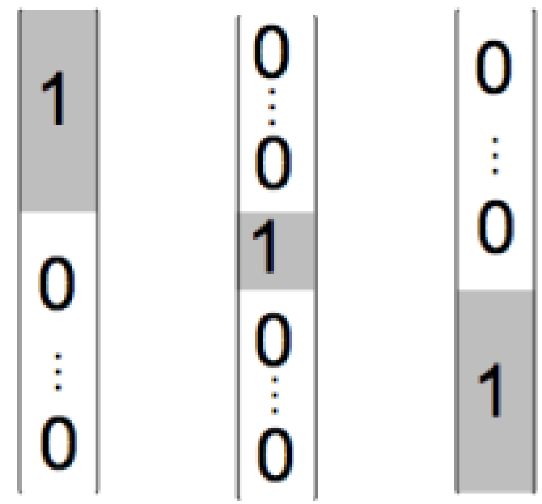


normalized
affinity matrix *L*



only for display: *L*
made block-diagonal
by sorting data points

matrix *X*



First three eigenvectors
k=3

Each row of *X* represents
one data point
→ points cluster at coor-
dinate axes in 3D space

Ex.: Eigenvectors to a block-diagonal Matrix

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \boxed{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

← No eigenvector
(vector gets rotated)

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \boxed{\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}} = \boxed{\begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}}$$

← Eigenvector!
(vector gets stretched only)

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \boxed{\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

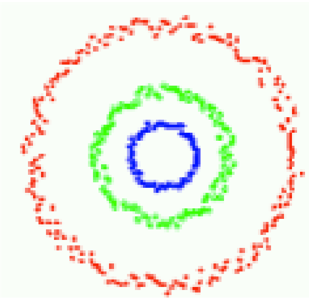
← No eigenvector
(vector gets rotated)

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \boxed{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} = \boxed{\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}$$

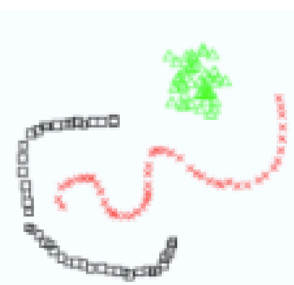
← Eigenvector!

Spectral Clustering Algorithm

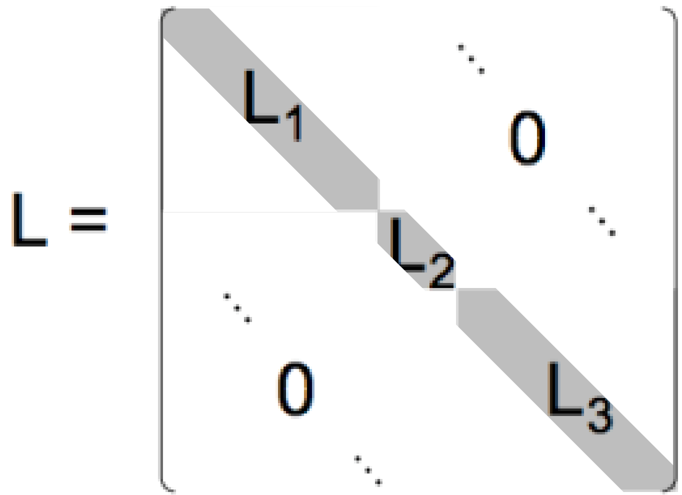
data



OR

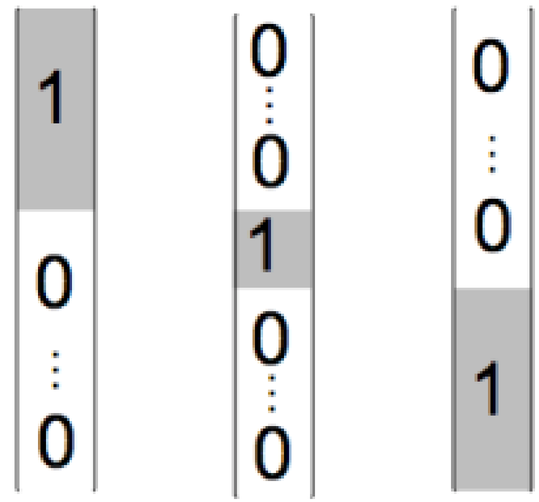


normalized
affinity matrix *L*



only for display: *L* made block-diagonal by sorting data points

matrix *X*



First three eigenvectors
k=3

Each row of *X* represents one data point
→ points cluster at coordinate axes in 3D space

Spectral Clustering Algorithm

- *For normalisation,* from X form the matrix $Y \in R^{nxk}$
  - Renormalize each of X's rows (it has n rows) to have unit length

$$Y_{ij} = X_{ij} / (\sum_j X_{ij}{}^2)^{1/2}$$

- Treat each row of Y as a point in R^k
  - all points will lie on a unit sphere around the origin
  - they will form clusters at ~90° to each other

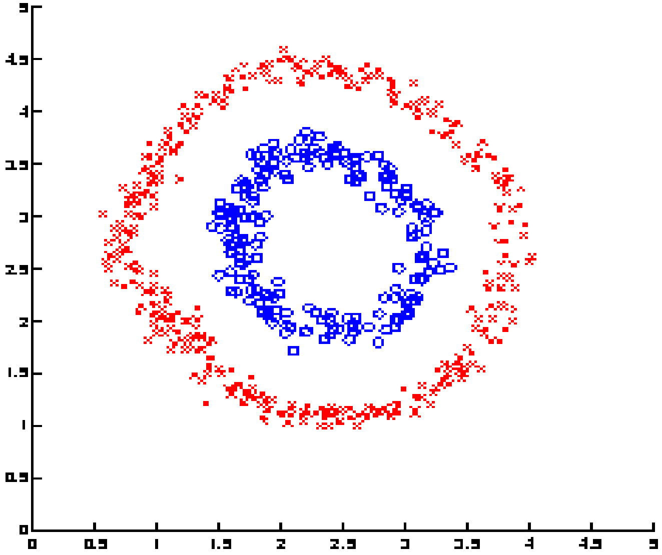
Spectral Clustering Algorithm

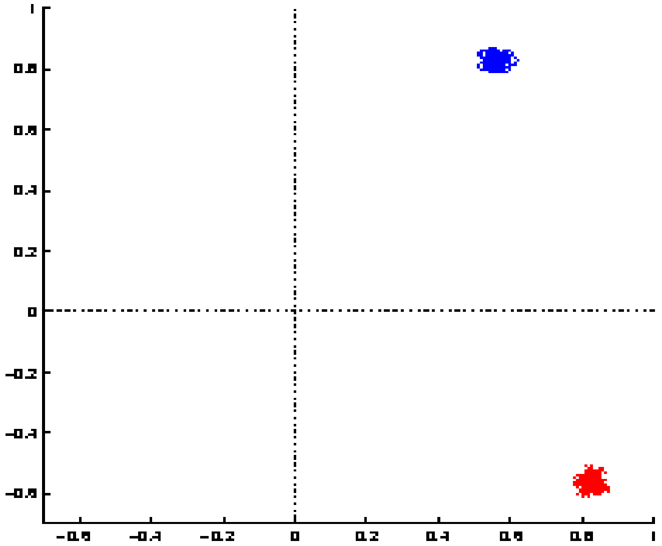
- Cluster into *k* clusters, e.g. via k-means
- Final cluster assignment
  - Assign point s_i to cluster *j*

    iff row *i* of *Y* was assigned to cluster *j*

Results

Spectral Clustering

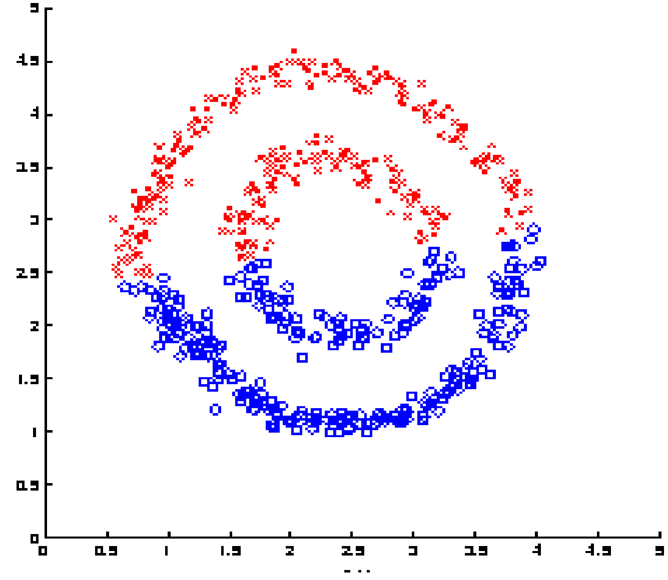




k=2

rows of Y

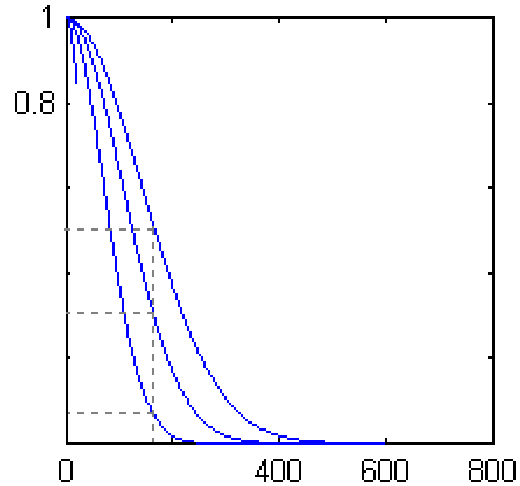
Comparison: k-Means
(directly on the data)



Spectral Clustering: Choice of σ

$$A_{ij} = e^{-(s_i - s_j)^2 / 2\sigma^2} \quad i \neq j$$

$$A_{ii} = 0$$

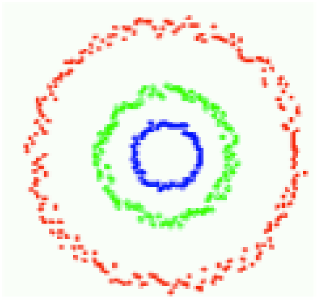


"closer" vertices get larger weight

- search over σ
  - pick the value that – after clustering *Y*'s rows – yields the tightest (smallest distortion) clusters

- Affinity metrics for connected graphs:
  - A_{ij} = 1, if nodes *i* and *j* are connected, else A_{ij} = 0
  - or: a function of geodesic distance

Spectral Clustering: Choice of *k*

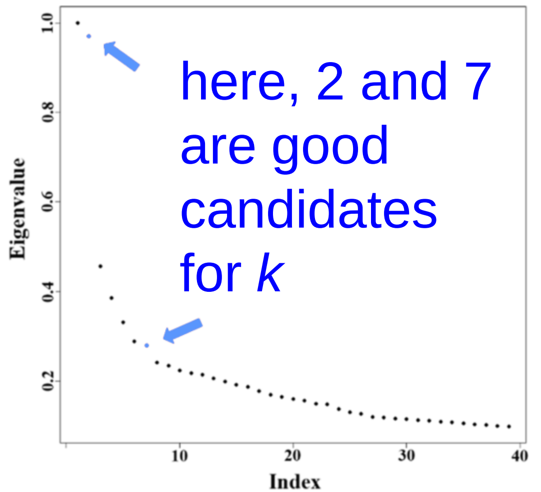


Is the Davies-Bouldin index a good measure for these kinds of clusters?

- No! It compares distances of points to mean versus distances between means (in this example, means are all same!)

Use *eigengap* heuristic to find the optimal *k*:

- Sort the eigenvalues (spectrum) of the normalized affinity matrix, and plot them according to that order
- Search for steps in that function:
  k is good if the k^{th} eigenvalue is large and eigenvalue $k+1$ drops off a lot

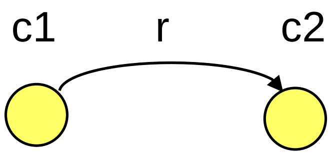


Bottom Figure: Talebi et al. (2019) Towards Geostatistical Learning for the Geosciences: A Case Study in Improving the Spatial Awareness of Spectral Clustering

Overview

- Spectral Clustering
- Semantic Networks
- Bayesian Belief Networks

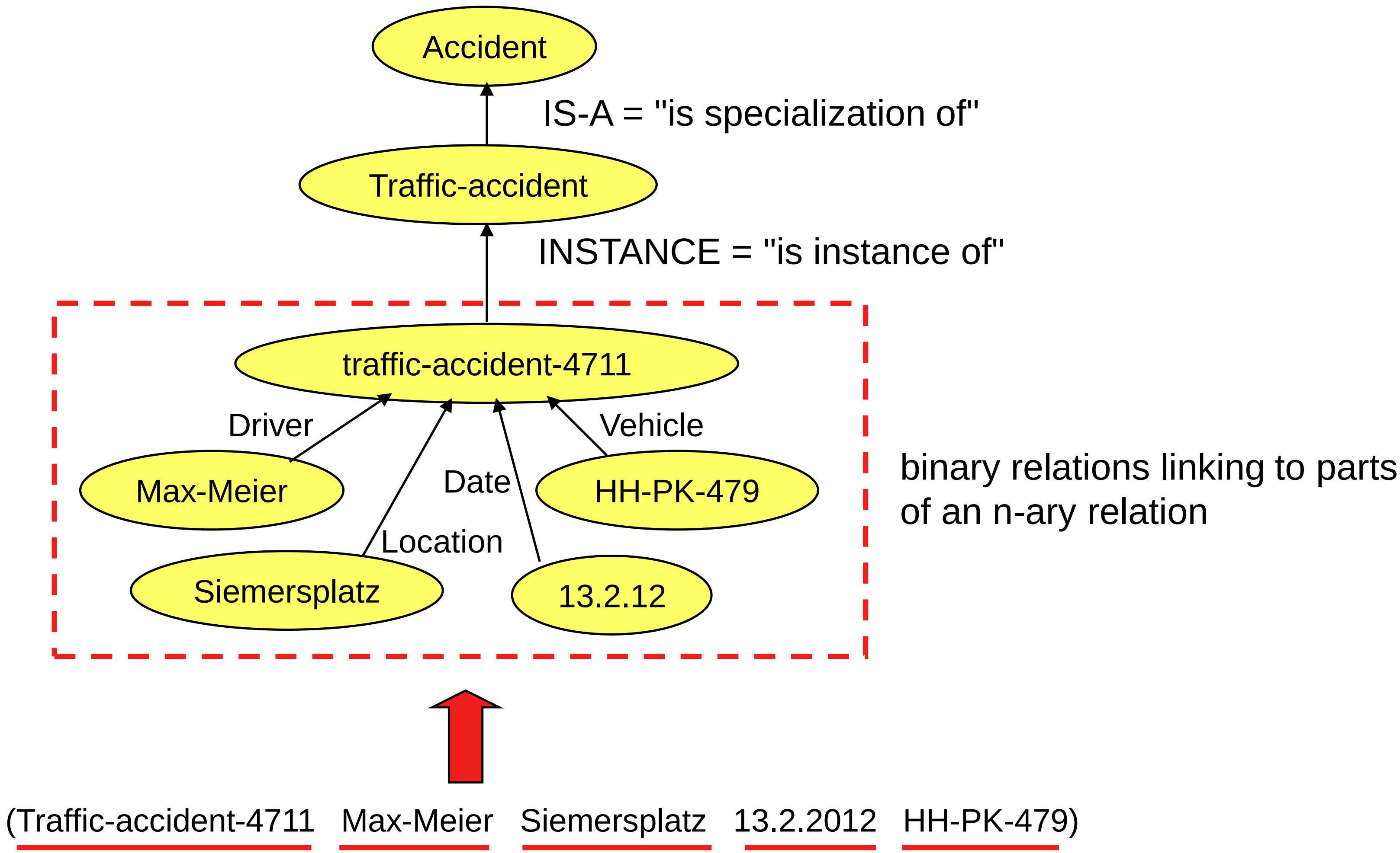
Semantic Networks



- Graphical representation of concepts and relations:
  - labeled nodes (vertices) = concepts
  - directed labeled links (edges) = binary relations
  - E.g.: $\underbrace{\text{dog}}_{c1}\ \underbrace{\text{is a}}_{r}\ \underbrace{\text{animal}}_{c2}$

- Where is the semantics?
  - Are there any types of ***nodes*** and types of ***links*** that are valid in general, independent of a particular domain?
  - Is there any ***structuring rule*** which is valid in general, independent of a particular domain?
  - Are there ***generally valid inference procedures*** to derive knowledge which is not explicitly stated?

Basic Relations in Semantic Networks

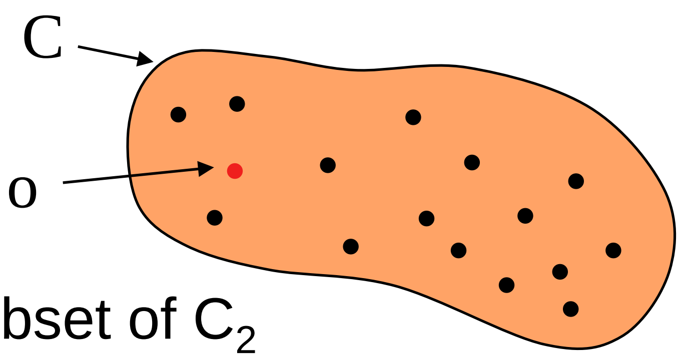


Concepts and Individuals

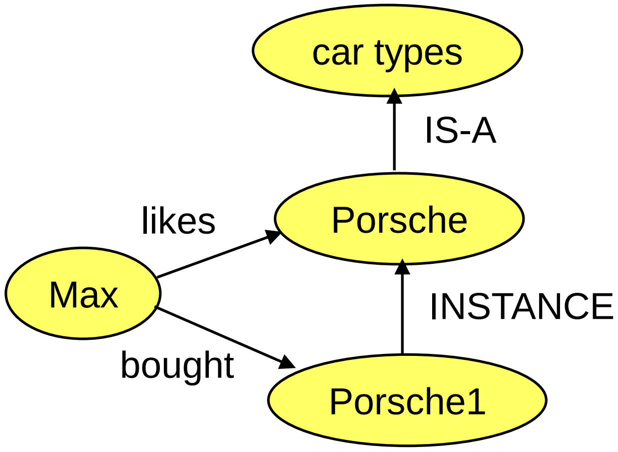
Nodes of a Semantic Network describe concepts and individuals.

A concept denotes a ***set of objects***.

An individual denotes a ***single object***.



C_1 *IS-A* C_2	specifies that C_1 is a subset of C_2
o INSTANCE C	specifies that o is a member of C

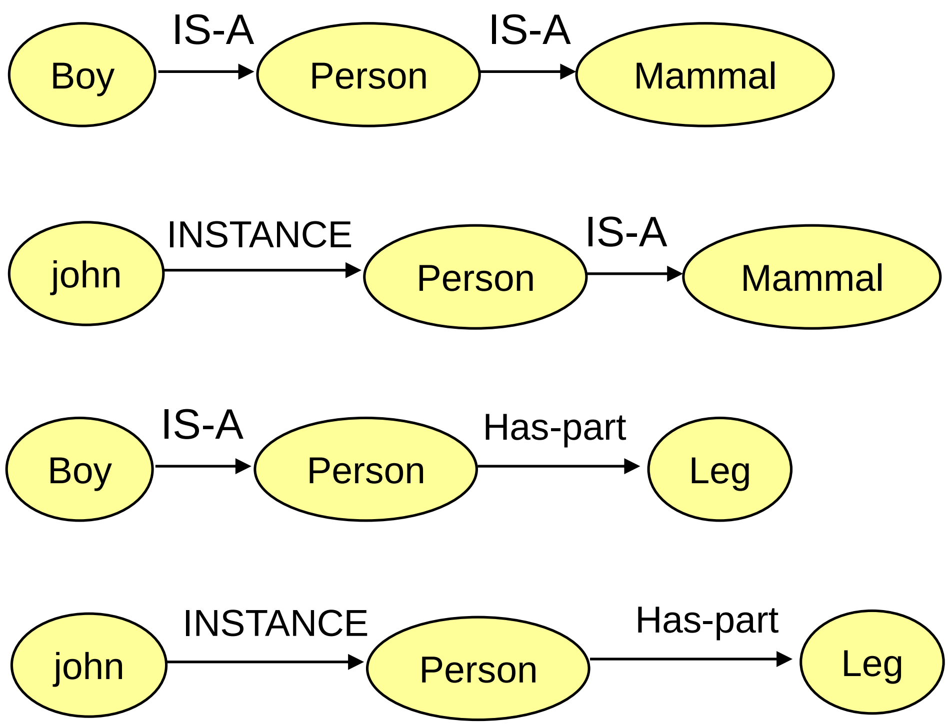


A node may represent both, an individual or a concept. Example:

Max likes a Porsche.
Max bought a Porsche
at the car dealer.

Inferences in Semantic Networks

Examples



Rules

C_1 IS-A C_2
C_2 IS-A C_3 $\Rightarrow$ C_1 IS-A C_3

c INSTANCE C_1
C_1 IS-A C_2 $\Rightarrow$ c INSTANCE C_2

C_1 IS-A C_2
C_2 Rel C_3 $\Rightarrow$ C_1 Rel C_3

c INSTANCE C_2
C_2 Rel C_3 $\Rightarrow$ c Rel C_3

Special Semantics for Special Relations

- Special relations may support special inferences.

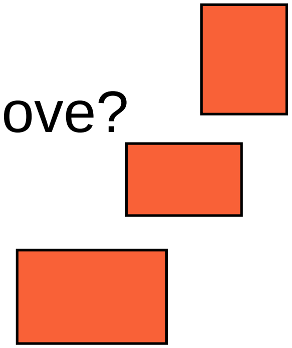
- **Examples**:

  $\text{Above}(a, b) \wedge \text{Above}(b, c) \Rightarrow \text{Above}(a, c)$

  $\text{Has-part}(a, b) \wedge \text{Has-Part}(b, c) \Rightarrow \text{Has-Part}(a, c)$

  $\text{Left}(a, b) \Rightarrow \text{Right}(b, a)$

Above?



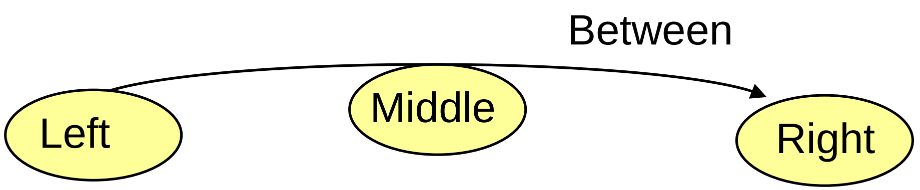
- The rules for inferences may change from domain to domain, hence they must be explicitly stated.

  ⇨ "axiomatizing a domain"

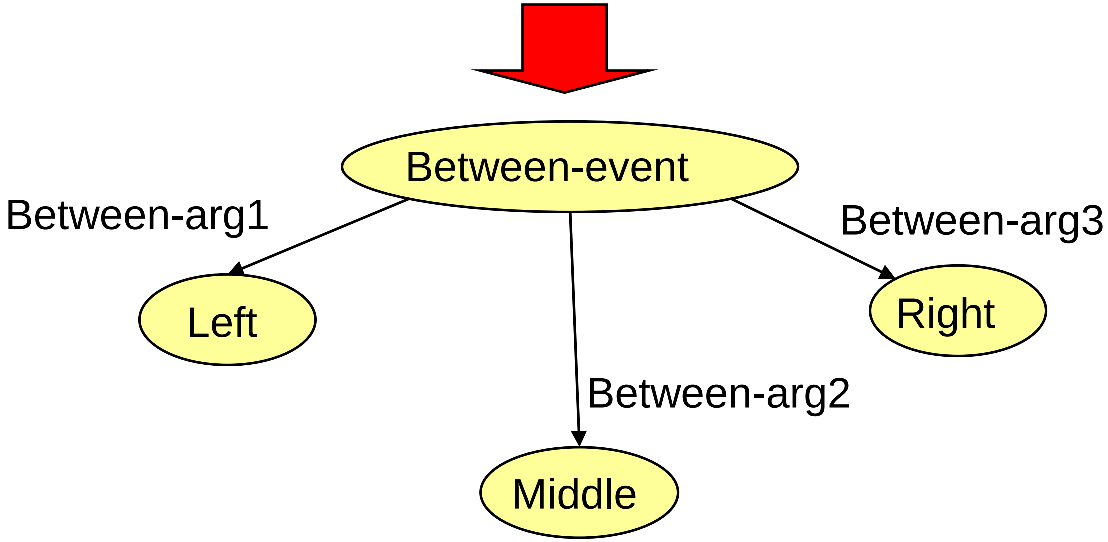
- Spatial reasoning, temporal reasoning are disciplines dealing with axiomatizations of spatial, part-of- and temporal relationships.

N-ary Relations in Semantic Networks

- Semantic Networks allow the representation of binary relations.
- Any N-ary relation can be represented by ***multiple binary relations***
- **Example**:

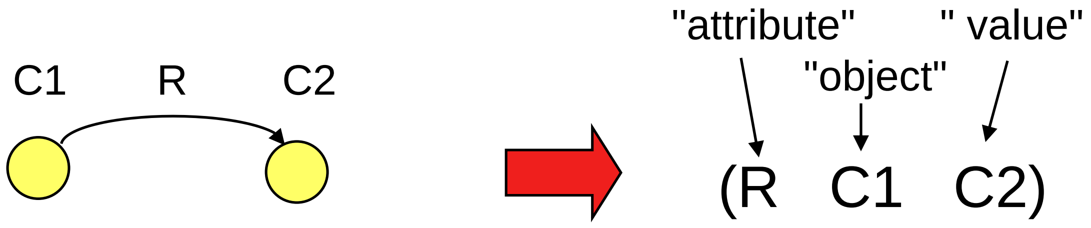


Between(Left, Middle, Right)



Attribute-Object-Value Triplets

- In knowledge representation- and programming languages, a Semantic Network can be represented by a set of triplets:



- The accident example:

  (is-a traffic-accident accident)

  (instance traffic-accident-4711 traffic-accident)

  (driver traffic-accident-4711 Max-Meier)

  (location traffic-accident-4711 Siemersplatz)

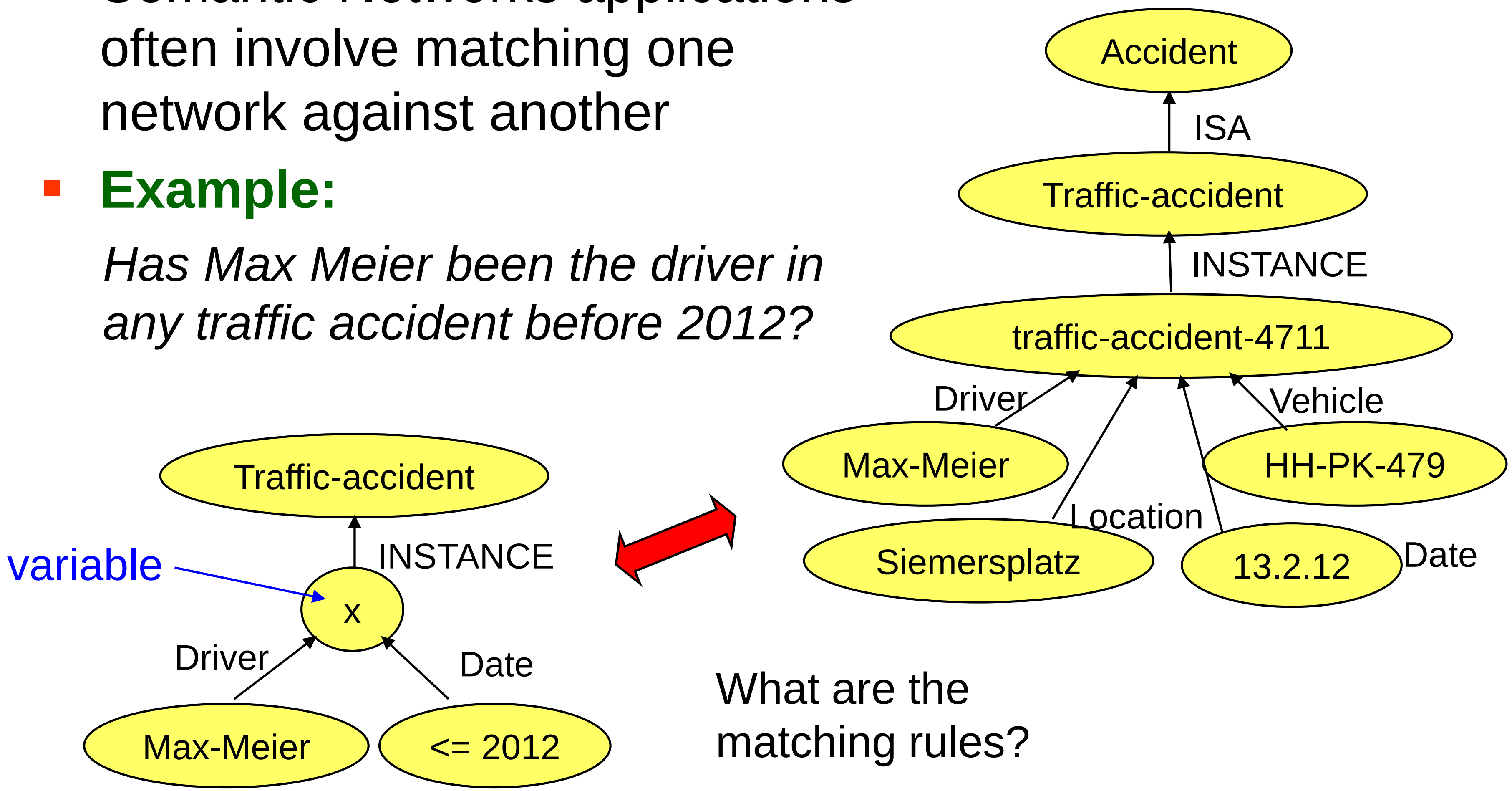
  (date traffic-accident-4711 13.2.12)

  (vehicle traffic-accident-4711 HH-PK-479)

Matching Relational Structures

- Semantic Networks applications often involve matching one network against another
- **Example:**

  Has Max Meier been the driver in any traffic accident before 2012?

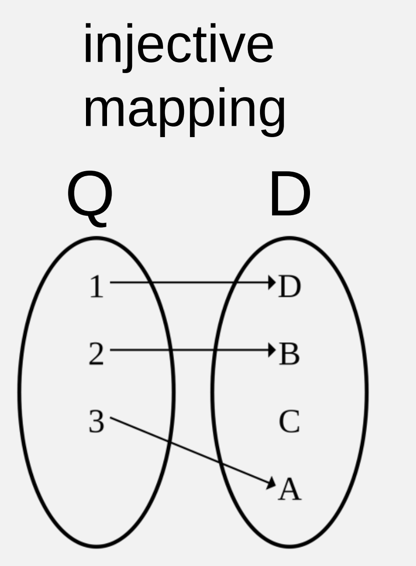


What are the matching rules?

Semantic Network (SN) Queries

- A SN query is a description of desired query responses in terms of a SN using an extended concept language.
- Typical concept language extensions:

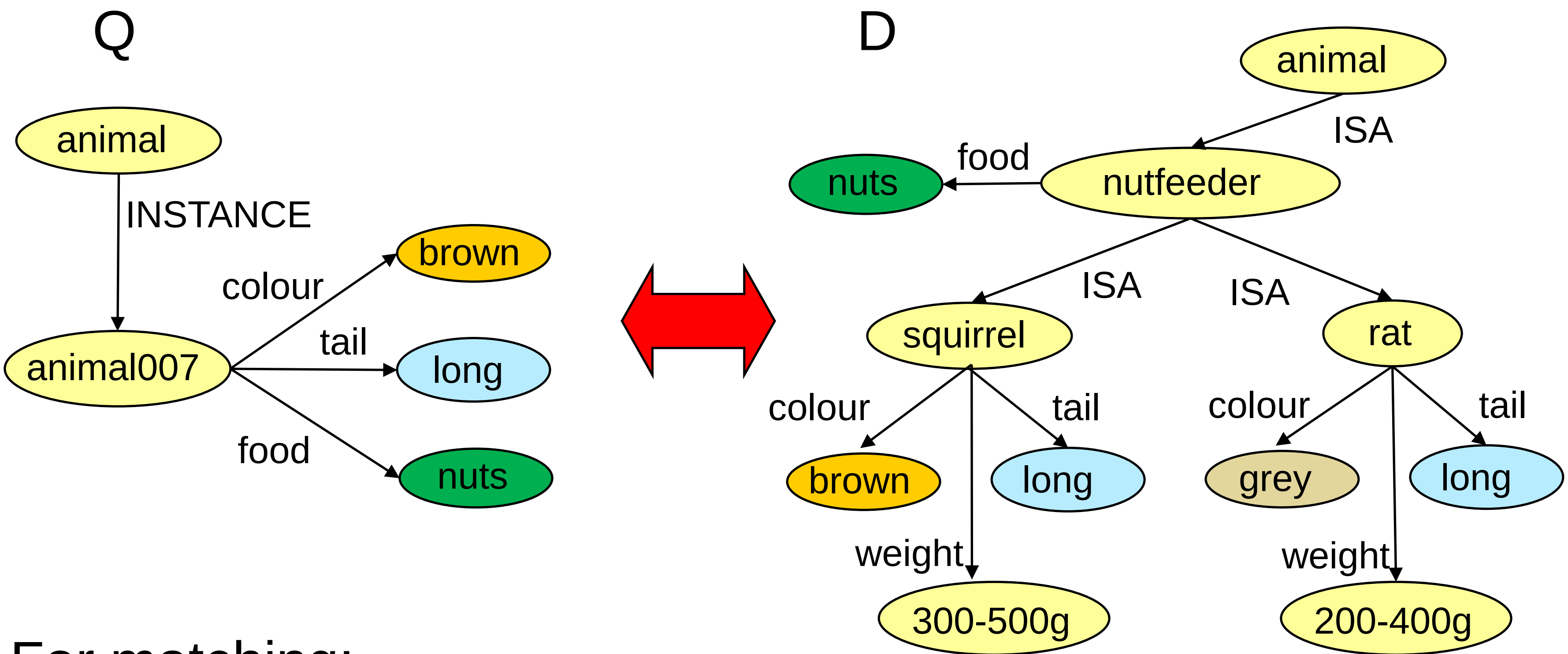
x	individual variable
X	concept variable
{a, b, c}	set of individuals
<= 2012	predicate over a concrete domain individual



Matching rules:

A query Q matches a database D, if there is an *injective* mapping of all nodes and links in Q to nodes and links in D such that the corresponding nodes and links are compatible.

Object Classification by Relational Matching



For matching:

- exploit INSTANCE and ISA inheritance
- Class descriptions must be given in terms of sufficient conditions

→ Graphs are classified by query matching

Semantic Networks – Summary

- Complex problems can be expressed by graphs
- Intuitive graphical knowledge representation
- Classification and information retrieval by query matching
- Semantics of relations is well-defined for ISA and INSTANCE, but not clearly defined in general
- Need for domain-specific inference rules (“axiomatizing”)
- Relations between relations cannot be expressed
- Generally, useful functions require additional formalisms such as rule-based inferences and new techniques from machine learning and automatic access, tagging, retrieval, pattern matching

Overview

- Spectral Clustering
- Semantic Networks
- Bayesian Belief Networks

Probabilistic Graphical Models

- Modelling of observations and their relationships
- Until now: semantic networks
- Reasoning over properties inherited from other instance(s)
  - *is*-*a*, *has*-*a* relationships

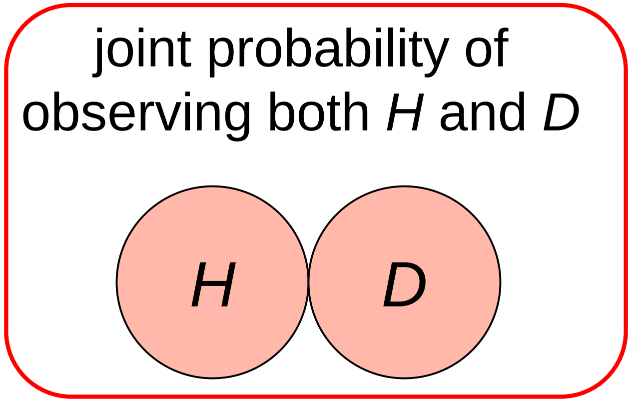
But how certain are we about the resulting statements?

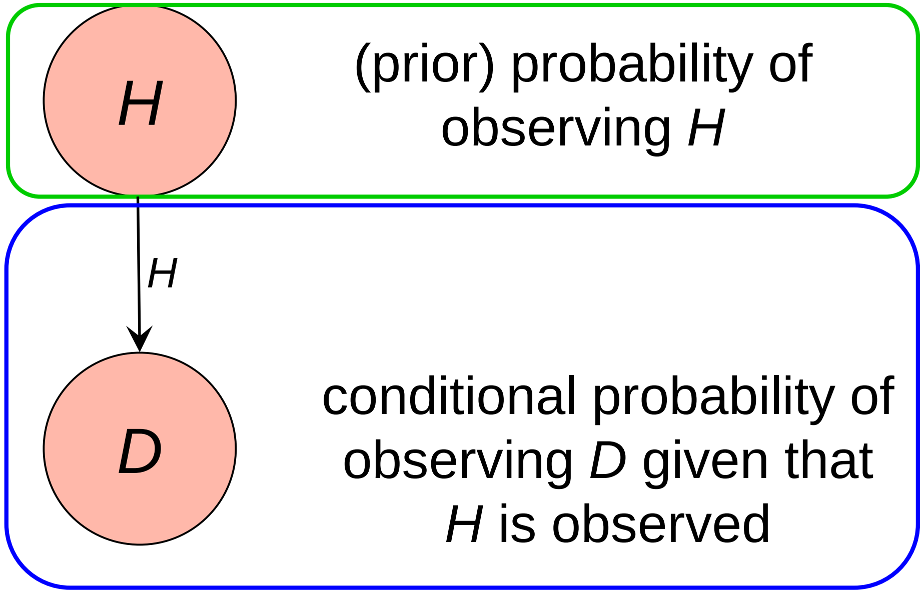
- Make inference under uncertainty
- Stochastics helps us with that
- Probabilistic Graphical Model can be
  directed or ***undirected***

Bayes Theorem

- Two stochastic events, *H* and *D*:

$$\left.\begin{aligned} \boxed{P(H,D)} &= \boxed{P(D \mid H)} \cdot \boxed{P(H)} \\ &= P(H \mid D) \cdot P(D) \end{aligned}\right\} \quad P(H \mid D) = \frac{P(D \mid H) \cdot P(H)}{P(D)}$$



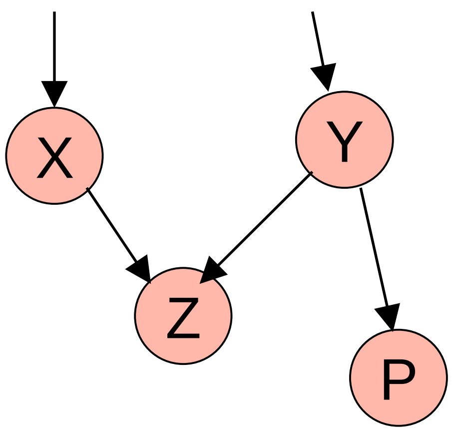


directed graphical model
(Bayes net)

Bayesian Belief Network

also known as probabilistic, or ***Bayesian network***. Components:

1. A *directed acyclic graph* (called a structure)
   - models *causal influence* relationships
   - represents *dependencies* among the variables
   - allows *class conditional independencies* between *subsets* of variables

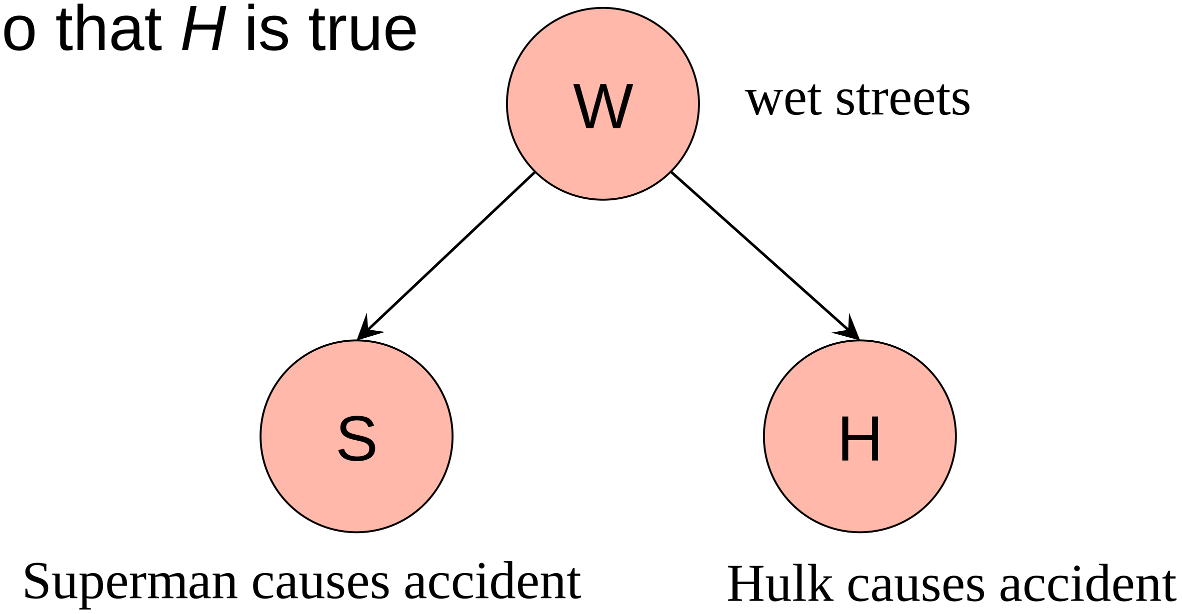


- Nodes: random variables
- Links: dependency
- X and Y are the parents of Z, and Y is the parent of P
- No dependency between Z and P
- Has no loops/cycles

2. A set of *conditional probability tables* (CPTs)
   - gives a specification of joint probability distribution

"Conditional Independence"

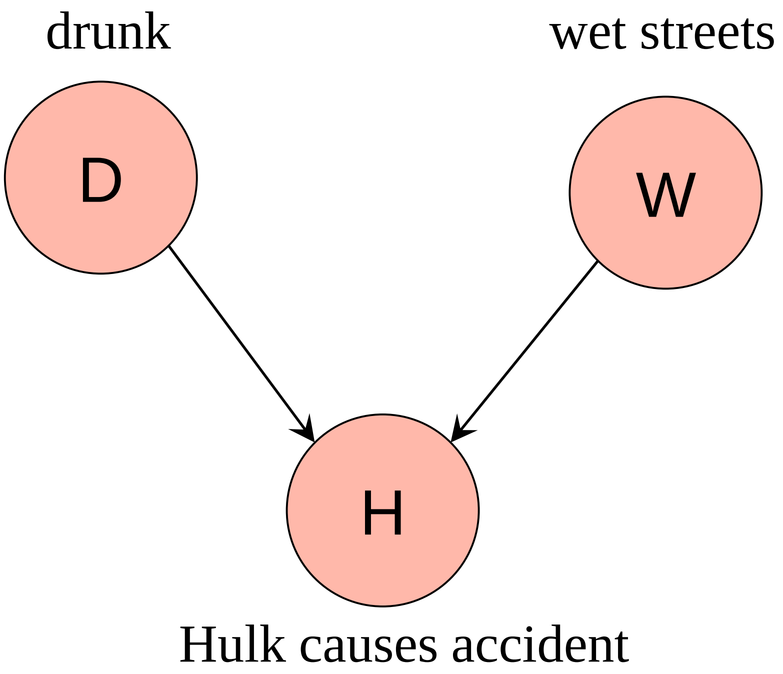
- *S* and *H* are both influenced by *W*
- If *S* is observed to be true, it is more likely that *W* is true, and then also that *H* is true



- If *W* is known (true or false), then *S* and *H* are independent
  $\rightarrow$ *S* and *H* are conditionally independent given *W*

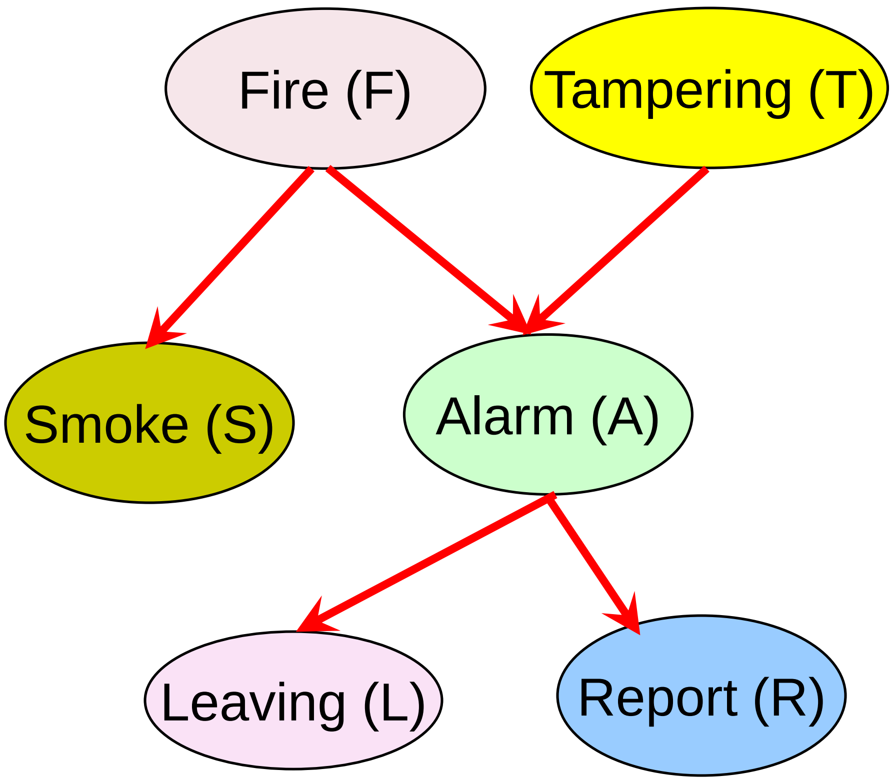
"Explaining away"

- *D* and *W* are independent
- both increase the likelihood of *H* to be true



- If *H* is observed, it is more likely that *D* or/and *W* are true
- If also *D* is observed to be true, then likelihood of *W* being true reduces again
  $\longrightarrow$ it is "explained away"

A Bayesian Network and Some of Its CPTs



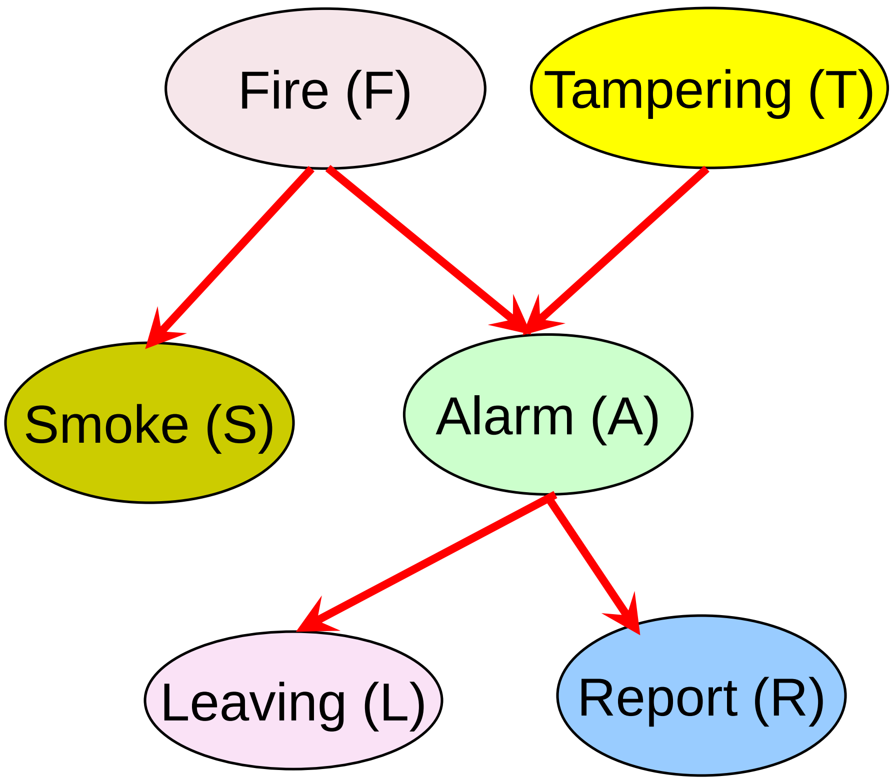
CPT: Conditional Probability Tables

Fire	Smoke	P(S|F)
True	True	.9
False	True	.01

Fire	Tampering	Alarm	P(A|F,T)
True	True	True	.5
True	False	True	.99
False	True	True	.85
False	False	True	.0001

CPT shows the conditional probability for each possible combination of its parents

A Bayesian Network and Some of Its CPTs



The joint probability *P(S,F,A,L,R,T)* is over *O(2^6)* states.

It would require (too) many observations to quantify the probabilities of all combinations.

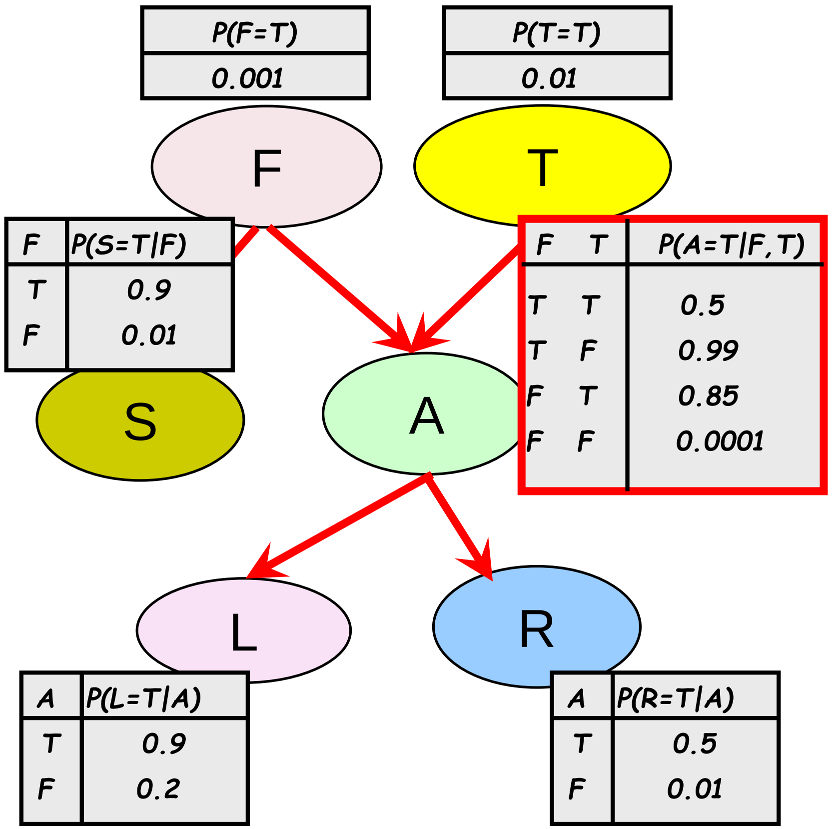
Bayes network: can compute the joint probability, while knowing only the conditional probabilities.

Savings arise due to missing links that denote no direct dependencies.

Derivation of the probability of a particular combination of values of *X*, from CPT:

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$$

A Bayesian Network and Some of Its CPTs



P(F=T)
0.001

P(T=T)
0.01

F	P(S=T|F)
T	0.9
F	0.01

F	T	P(A=T|F,T)
T	T	0.5
T	F	0.99
F	T	0.85
F	F	0.0001

A	P(L=T|A)
T	0.9
F	0.2

A	P(R=T|A)
T	0.5
F	0.01

has $O(2^6)$ combinations

$P(S,F,A,L,R,T)$

product rule

$= P(T)\, P(S,F,A,L,R|T)$

F and S are independent of T

$= P(T)\, P(S,F|\cancel{T})\, P(A,L,R|S,F,T)$

A,L,R don't depend on S

product rule

$= P(T)\, P(F)\, P(S|F)\, P(\underline{A,L,R}|F,T)$

L and R are conditionally independent of F and T given A

$= P(T)\, P(F)\, P(S|F)\, P(\underline{A}|F,T)\, P(\underline{L,R|A\cancel{,F,T}})$

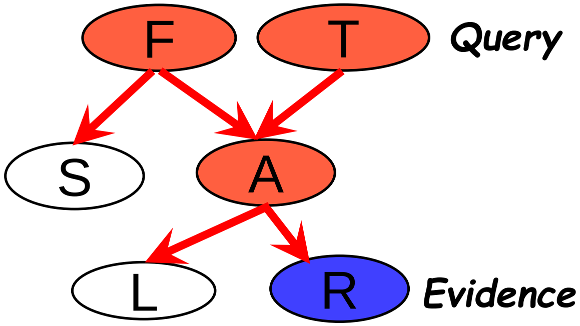
L and R are independent given A

$= P(T)\, P(F)\, P(S|F)\, P(A|F,T)\, P(L|A)\, P(R|A)$

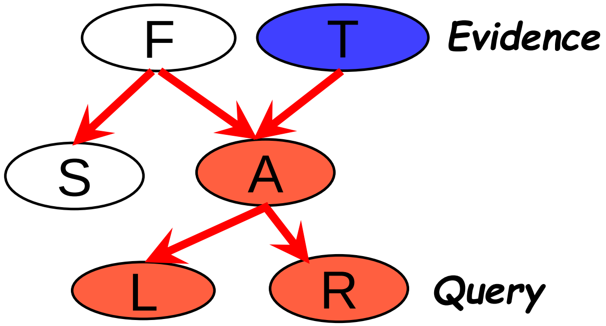
has $O(2^2)$ combinations only

Types of Reasoning in Bayesian Networks

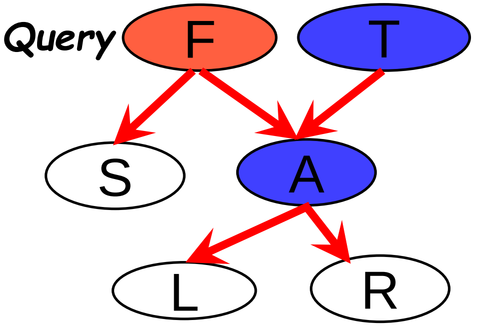
- Diagnostic
  - From symptoms to causes, e.g., doctor infers diseases from symptoms. Reasoning occurs in opposite direction to network arcs.



- Predictive
  - Reasoning from new information about causes to new beliefs about effects, follows the directions of the networks arcs.



- Inter-causal (explaining away)
  - Mutual causes of a common effect. Initially, causes may be independent. But if a common effect is observed and we learn that one cause is true, then the other is less likely.



- Conditional Independence
  - Mutual effects of a common cause. If one effect is observed, the cause and hence the other effect is more likely. If we know the cause, then the effects become independent.

How Are Bayesian Networks Constructed?

- ***Subjective construction***: Identification of (direct) causal structure
  - People are quite good at identifying direct causes from a given set of variables & whether the set contains all relevant direct causes
  - *Markovian assumption*: Each variable becomes independent of its non-effects once its direct causes are known

    E.g., S ← F → A ← T, path S → A is blocked once we know F → A
  - HMM (Hidden Markov Model): often used to model dynamic systems whose states are not observable, yet their outputs are
- ***Synthesis*** from other specifications
  - E.g., from a formal system design: block diagrams & info flow
- ***Learning from data***
  - E.g., from medical records or student admission record
  - Learn parameters given its structure or learn both structure and params
  - Maximum likelihood principle: favors Bayesian networks that maximize the probability of observing the given data set

Training Bayesian Networks

- Scenario 1: Given the network structure and all variables observable:
  → *compute only the CPT entries*
- Scenario 2: Network structure known, some variables hidden:
  → *gradient descent* (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment
  - Weights are updated at each iteration & converge to local optimum
- Scenario 3: Network structure unknown, all variables observable:
  → search through the model space to *reconstruct network topology*

[D. Heckerman. A Tutorial on Learning with Bayesian Networks. In *Learning in Graphical Models,* M. Jordan, ed. MIT Press, 1999]

Summary

- Spectral clustering
  - for graphs, or for any points with distances in between
- Semantic networks
  - Graphical models for knowledge representation
  - Domains need axiomatizing
- Probabilistic reasoning
  - Bayesian Belief Networks efficiently compute joint probabilities
  - benefit from direct independencies among the variables