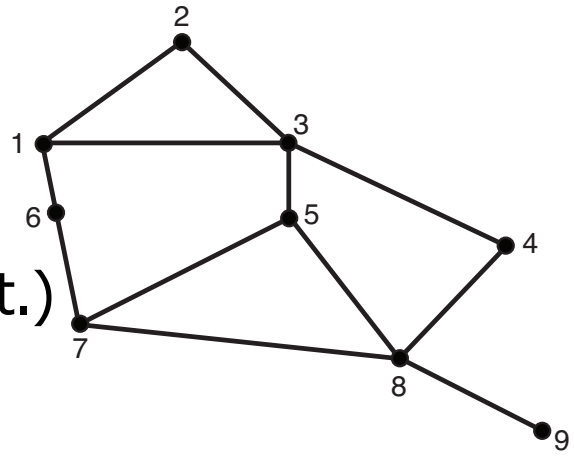
Learning and Inference for Graphical and Hierarchical Models: A Personal Journey

Alan S. Willsky
willsky@mit.edu
http://lids.mit.edu
http://ssg.mit.edu

May 2013

Undirected graphical models



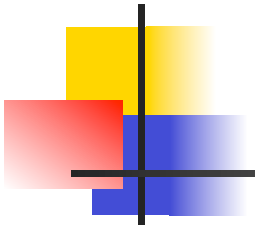
- $\boldsymbol{G} = (V, E)$ (V=vertices; $E \subset V \times V$ = edges)
- Markovianity on $\boldsymbol{G}$
  - Hammersley-Clifford (NASC for positive dist.)
    - $\mathbf{C}$ = Set of all cliques in $\boldsymbol{G}$

$$p(x) = \frac{1}{Z} \exp\left\{ \sum_{C \in \mathbf{C}} \varphi_C(x_C) \right\}$$

    - φ_c = Clique potential Z = partition function
- Pairwise graphical models

$$p(x) = \frac{1}{Z} \exp\left\{ \sum_{s \in V} \varphi_s(x_s) + \sum_{(s,t) \in E} \varphi_{st}(x_s, x_t) \right\}$$

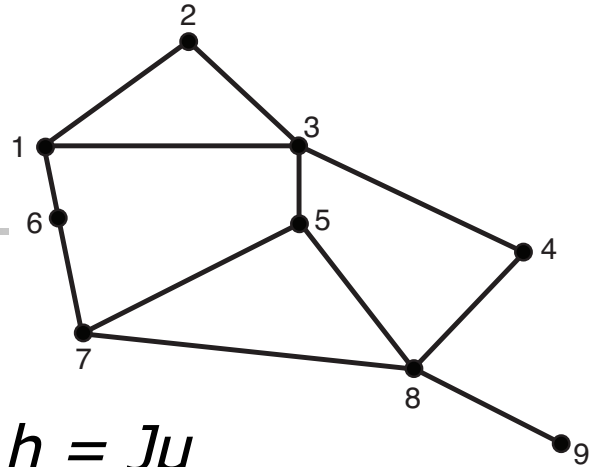
$$p(x) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{st}(x_s, x_t)$$



Directing/undirecting the graphs?

- Undirected models: Factors are ***not*** typically probabilities
  - Although they ***can*** be for cycle-free graphs (see BP), i.e., ***trees***
- Directed models: Specify in terms of transition probabilities (parents to children)
- Directed to undirected: Easy (after moralization)
- Undirected to directed: Hard (and often a mistake) unless the graph is a tree (see BP)

Gaussian models



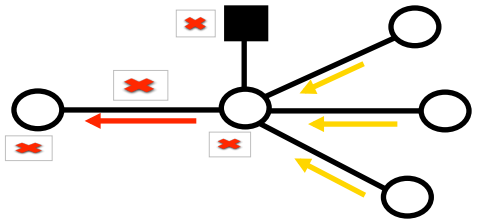
- $X \sim N(\mu, P)$ or $N^{-1}(h, J)$, with $J = P^{-1}$ and $h = J\mu$
- The sparsity structure of J determines graph structure
  - I.e., $J_{st} = 0$ if (s, t) is not an edge
- Directed model (0-mean for simplicity):

$$AX = w$$

  - $W \sim N(0, I)$
  - A – Lower triangular
- $A = J^{1/2}$ ➔ In general we get lots of ***fill***, unless the graph structure is a tree
- And it's even more complicated for non-Gaussian models, as higher-order cliques are introduced

Belief Propagation: Message passing for pairwise models on ***trees***

- Fixed point equations for likelihoods from disjoint parts of the graph:



$$m_{ts}(x_s) = \alpha \int_{x_t} \psi_{s,t}(x_s, x_t)\psi_t(x_t) \prod_{u \in \mathcal{N}(t)\setminus s} m_{ut}(x_t)dx_t$$

$$P_s(x_s) = \alpha\psi_s(x_s) \prod_{u \in \mathcal{N}(s)} m_{us}(x_s)$$

$$P_{st}(x_s, x_t) = \alpha\psi_{st}(x_s, x_t)\psi_s(x_s)\psi_t(x_t) \prod_{u \in \mathcal{N}(s)/t} m_{us}(x_s) \prod_{u \in \mathcal{N}(t)/s} m_{ut}(x_t)$$

BP on trees

- Gives factored form for distribution in terms of probability distributions

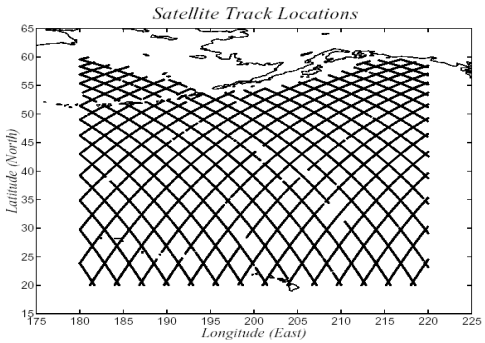
$$P(\{x_s | s \in V\}) = \prod_{s \in V} P_s(x_s) \prod_{(s,t) \in E} \frac{P_{st}(x_s, x_t)}{P_s(x_s) P_t(x_t)}$$

- Great flexibility in message-scheduling
  - Leaves-root-leaves = Rauch-Tung-Striebel
  - Completely parallel messaging, convergence in number of steps = diameter of the graph

Modeling of ***structure***: Four questions

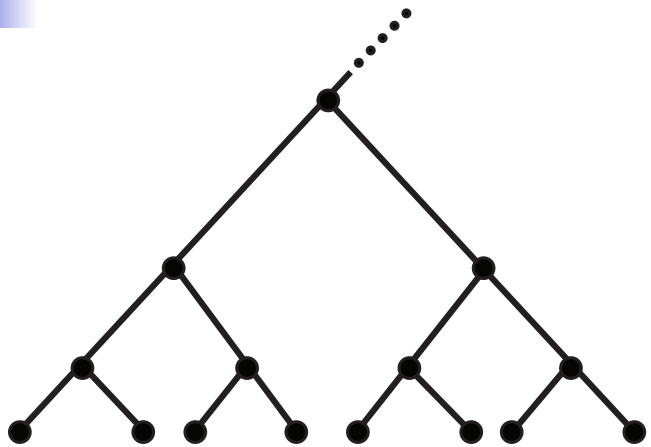
- What should the structure of the graph be?
- Which variables go where?
- What about adding hidden (unobserved) nodes (and variables)?
- What should the dimensions of the various variables be?

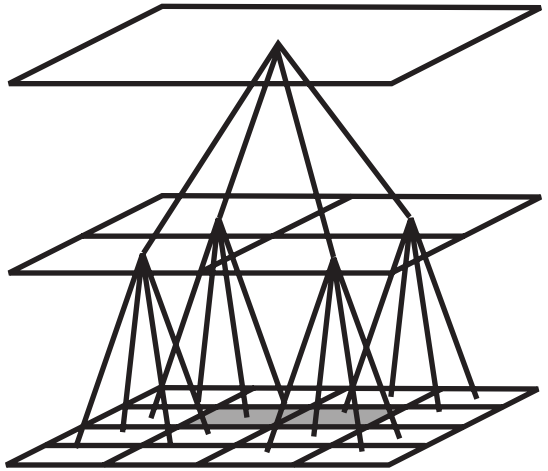
Our initial foray (with thanks to Michèle Basseville and Albert Benveniste): Multiresolution Models



- What can be multiresolution?
  - The ***phenomenon*** being modeled
  - The ***data*** that are collected
  - The ***objectives*** of statistical inference
  - The ***algorithms*** that result
- Some applications that motivated us (and others)
  - Oceanography
  - Groundwater hydrology
  - "Fractal priors" in regularization formulations in computer vision, mathematical physics, …
  - Texture discrimination
  - Helioseismology (?) …

Specifying MR models on trees





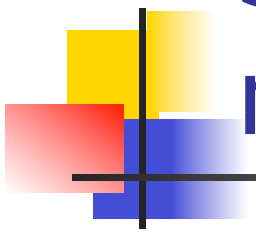
- MR ***synthesis***, leads, as with Markov chains, to thinking about ***directed trees***:
- E.g.: $x(s) = A(s)x(s\gamma) + w(s)$
  - E.g.: Midpoint deflection is such a model
- Note that the ***dimension*** of the variables comes into play
- But let's assume we pre-specify the tree structure

A control theorist's idea: **Internal models**

- Variables at coarser nodes are ***linear functionals*** of the finest-scale variables
  - Some of these may be measured or important quantities to be estimated
  - The rest are to be ***designed***
    - To approximate the condition for tree-Markovianity
    - To yield a model that is "close" to the true fine-scale statistics
  - Scale-recursive algebraic design
    - Criterion used: Canonical correlations or predictive efficiency
  - Alternate using midpoint deflection or wavelets**
- Confounding the control theorist
  - Internal models need not have minimal state dimension

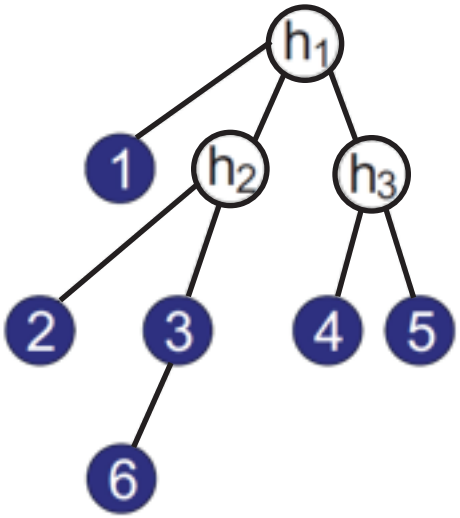
$$x(s) = x(s\overline{\gamma}) + w(s)$$

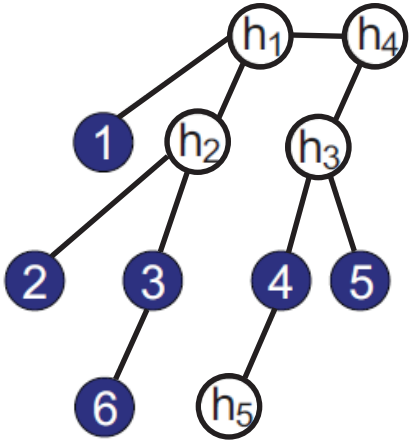
So, what if we want a tree but not necessarily a hierarchical one



- One approach: Construct the maximum likelihood tree given sample data (or the full second-order statistics)
- ***NOTE:*** This is quite different from what system theorists typically consider
  - There is no "state" to identify: All of the variables in the model we desire are observed
  - It is the ***index set*** that we get to play with
- Chow-Liu found a very efficient algorithm
  - Form graph with each observed variable at a different node
  - Edge weight between any two variables is their mutual information
  - Compute max-weight spanning tree
- What if we want hidden/latent "states"?

Reconstruction of a Latent Tree





- Reconstruct a latent tree using exact statistics (first) or samples (to follow) of observed nodes only
- Exact/consistent recovery of ***minimal*** latent trees
  - Each hidden node has at least 3 neighbors
  - Observed variables are neither perfectly dependent nor independent
- Other objectives:
  - Computational efficiency
  - Low sample complexity

Information Distance

- Gaussian distributions

$$d_{ij} := -\log |\rho_{ij}|$$

$$\rho_{ij} := \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_j)}}$$

- Discrete distributions

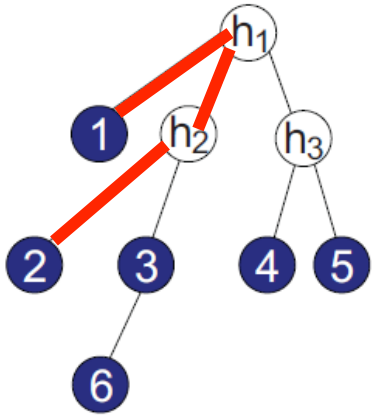
$$d_{ij} := -\log \frac{|\det \mathbf{J}^{ij}|}{\sqrt{\det \mathbf{M}^i \det \mathbf{M}^j}}$$

$\mathbf{J}^{ij}$ Joint probability matrix

$\mathbf{M}^i$ Marginal probability matrix (diagonal)

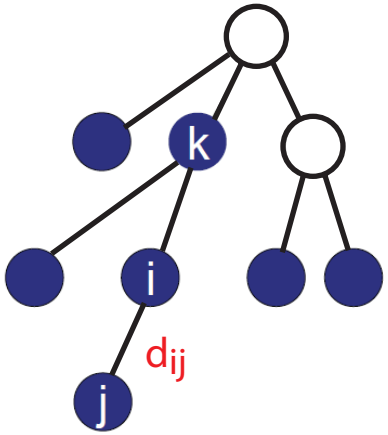
Additivity

$$d_{k,l} = \sum_{(i,j)\in \mathrm{Path}((k,l);E_p)} d_{i,j}$$



$$d_{12} = d_{1h_1} + d_{h_1h_2} + d_{2h_2}$$

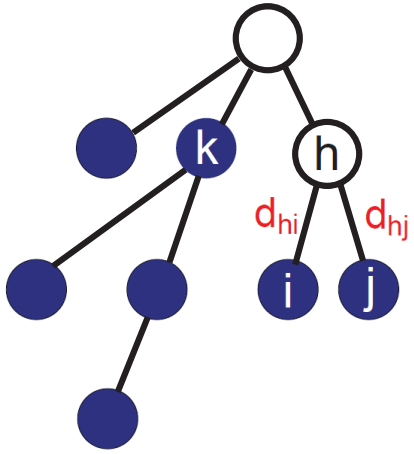
Testing Node Relationships



Node j – a leaf node Node i – parent of j

$\Leftrightarrow$ for all $k \neq i, j$ $d_{jk} - d_{ik} = d_{ij}$

Can identify (parent, leaf child) pair

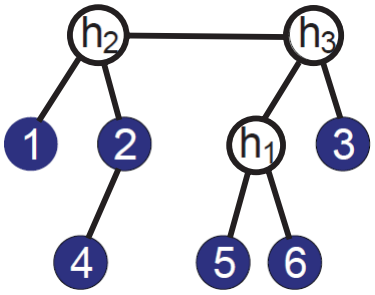


Node i and j – leaf nodes and share the same parent (sibling nodes)

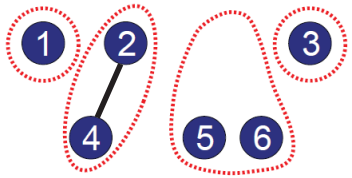
$\Leftrightarrow$ for all $k \neq i, j$ $d_{jk} - d_{ik} = d_{hj} - d_{hi}$

Can identify leaf-sibling pairs.

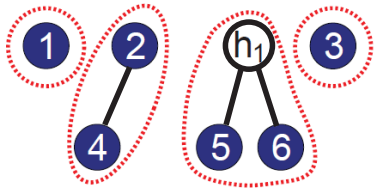
Recursive Grouping (exact statistics)



Step 1. Compute $d_{jk} - d_{ik}$ for all observed nodes (i, j, k).

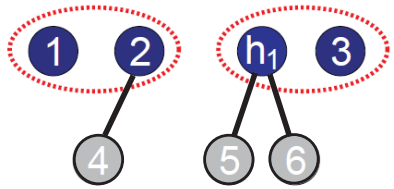


Step 2. Identify (parent, leaf child) or (leaf siblings) pairs.



Step 3. Introduce a hidden parent node for each sibling group without a parent.

Step 4. Compute the information distance for new hidden nodes. E.g.: $d_{5h_1} = \frac{1}{2}(d_{56} + d_{53} - d_{63})$



Step 5. Remove the identified child nodes and repeat Steps 2-4.

Recursive Grouping

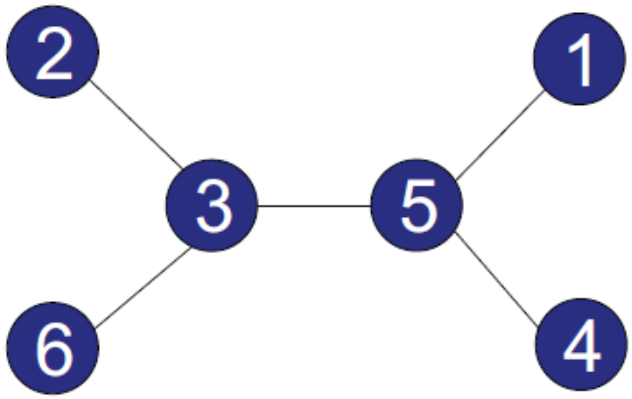
- Identifies a group of family nodes at each step.
- Introduces hidden nodes recursively.
- Correctly recovers all minimal latent trees.

- Computational complexity $O(\text{diam}(T)\, m^3)$.

Worst case $O(m^4)$

Chow-Liu Tree

$$\mathrm{MST}(V; \mathbf{D})$$



Minimum spanning tree of V

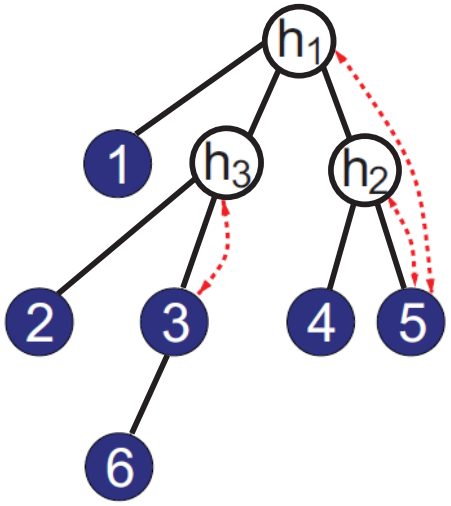
using D as edge weights

V = set of observed nodes

D = information distances

- Computational complexity $O(m^2 \log m)$

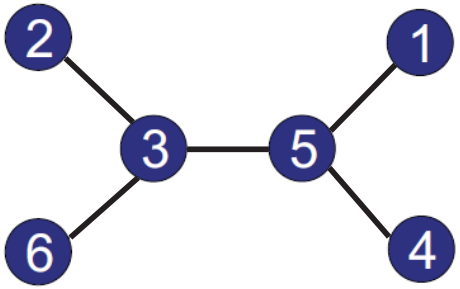
Surrogate Nodes and the C-L Tree



V = set of observed nodes

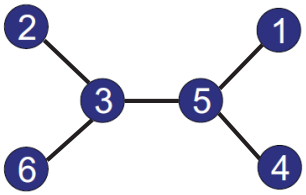
Surrogate node of i

$$\mathrm{Sg}(i) := \operatorname*{argmin}_{j \in V} d_{ij}$$

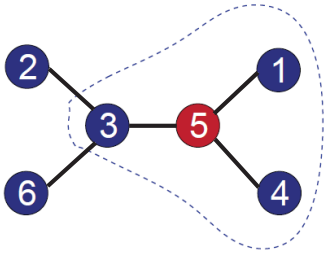


If (*i*, *j*) is an edge in the latent tree, then (Sg(*i*), Sg(*j*)) is an edge in the Chow-Liu tree

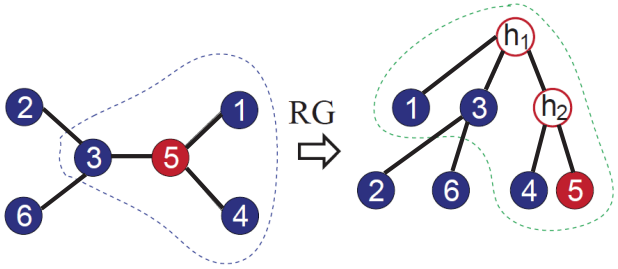
CLGrouping Algorithm



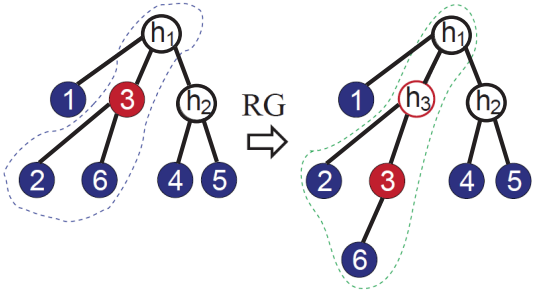
Step 1. Using information distances of observed nodes, construct MST(V; D). Identify the set of internal nodes.



Step 2. Select an internal node and its neighbors, and apply the recursive-grouping (RG) algorithm.



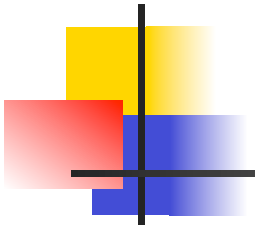
Step 3. Replace the output of RG with the sub-tree spanning the neighborhood.



Repeat Steps 2-3 until all internal nodes are operated on.

Computational complexity

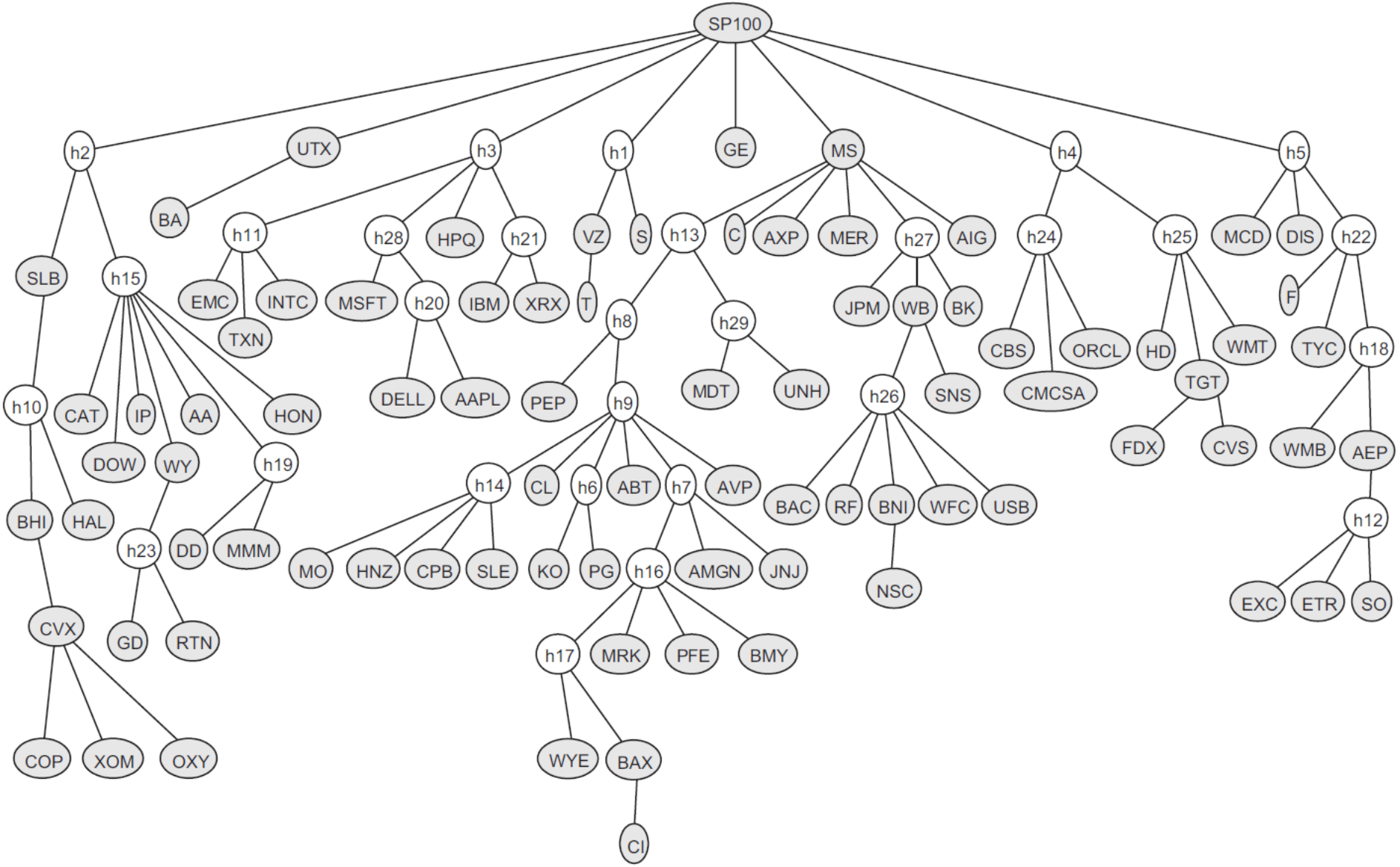
$O(m^2 \log m + (\#\text{internal nodes})\,(\text{maximum degree})^3)$

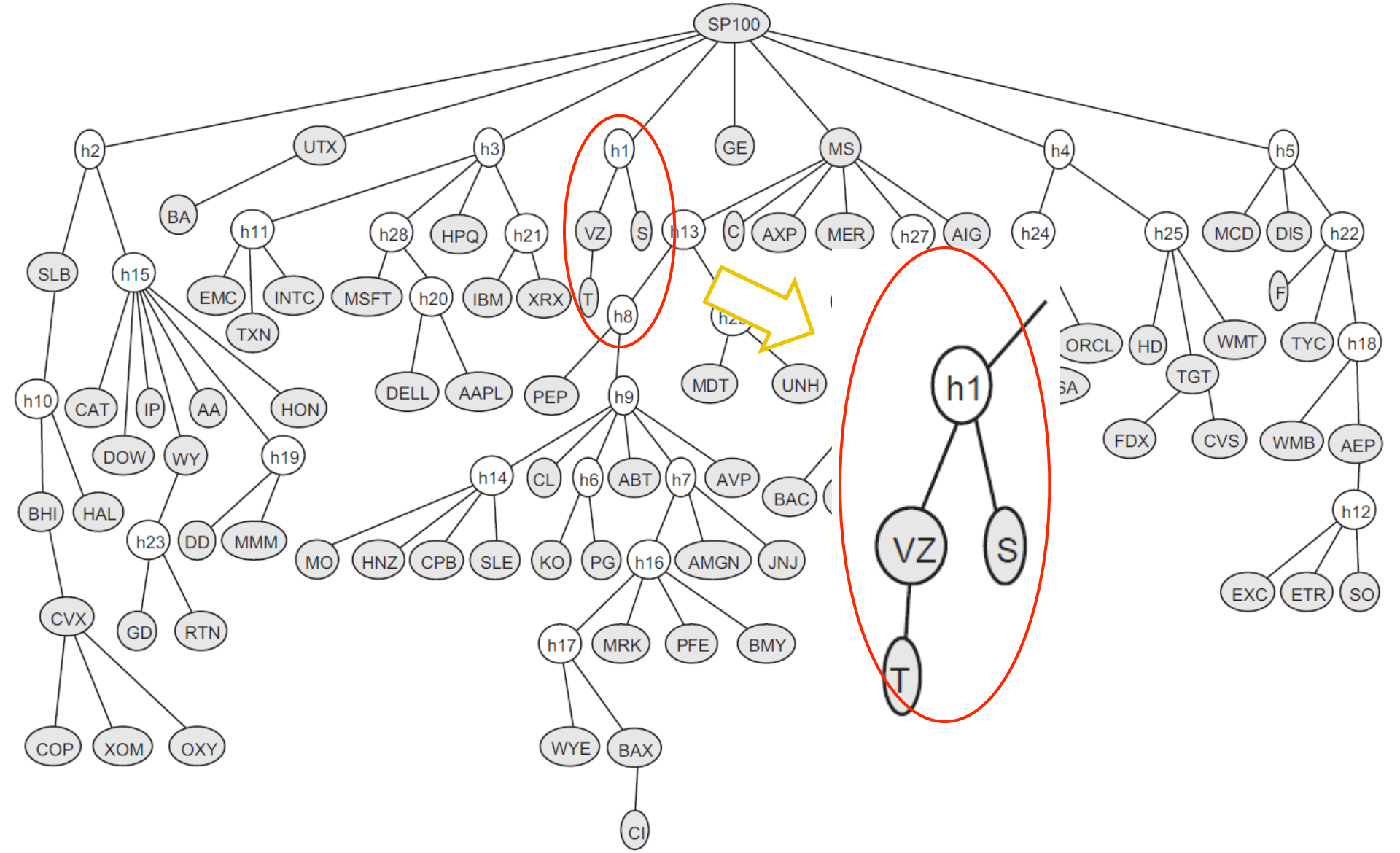


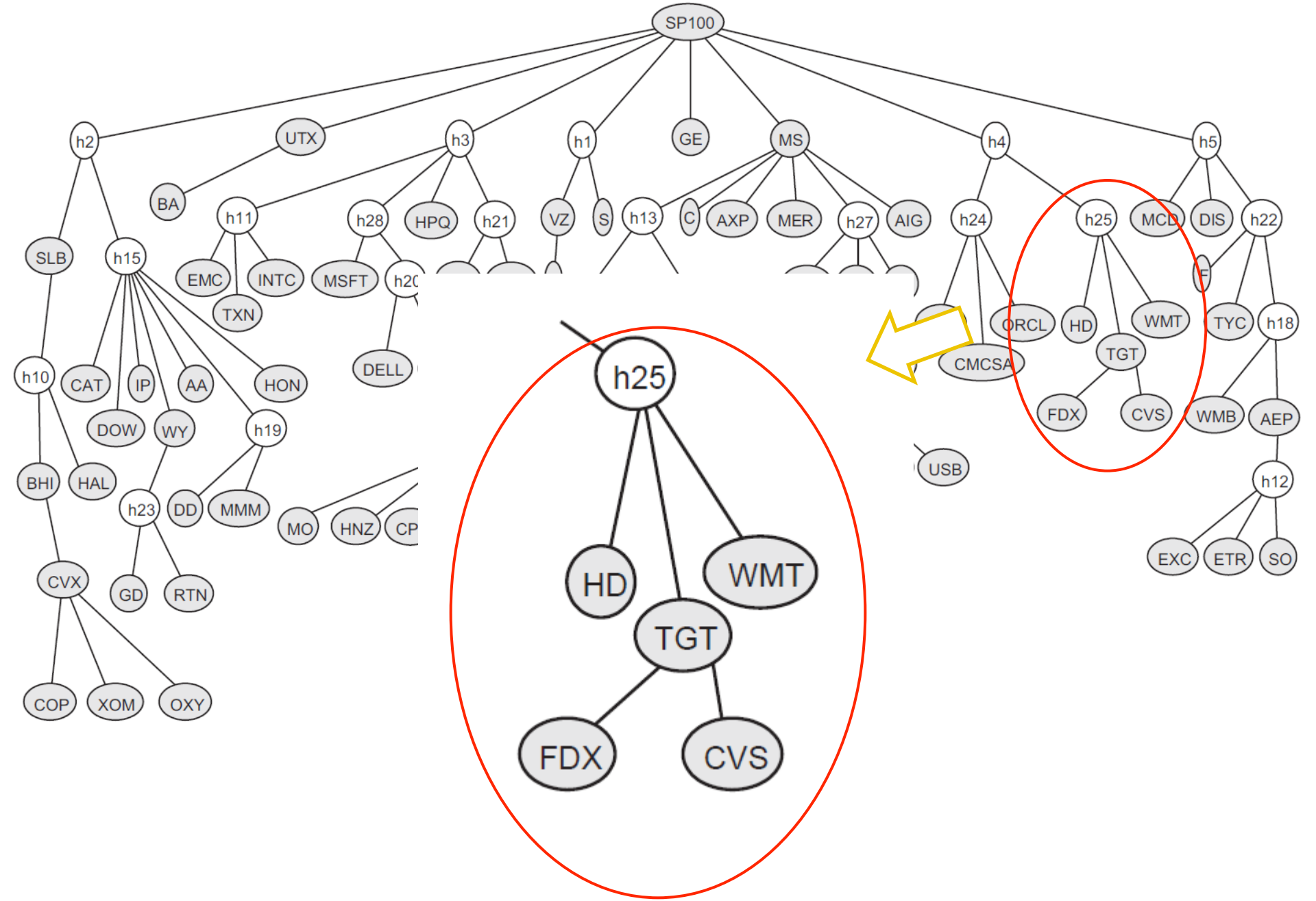
Sample-based Algorithms

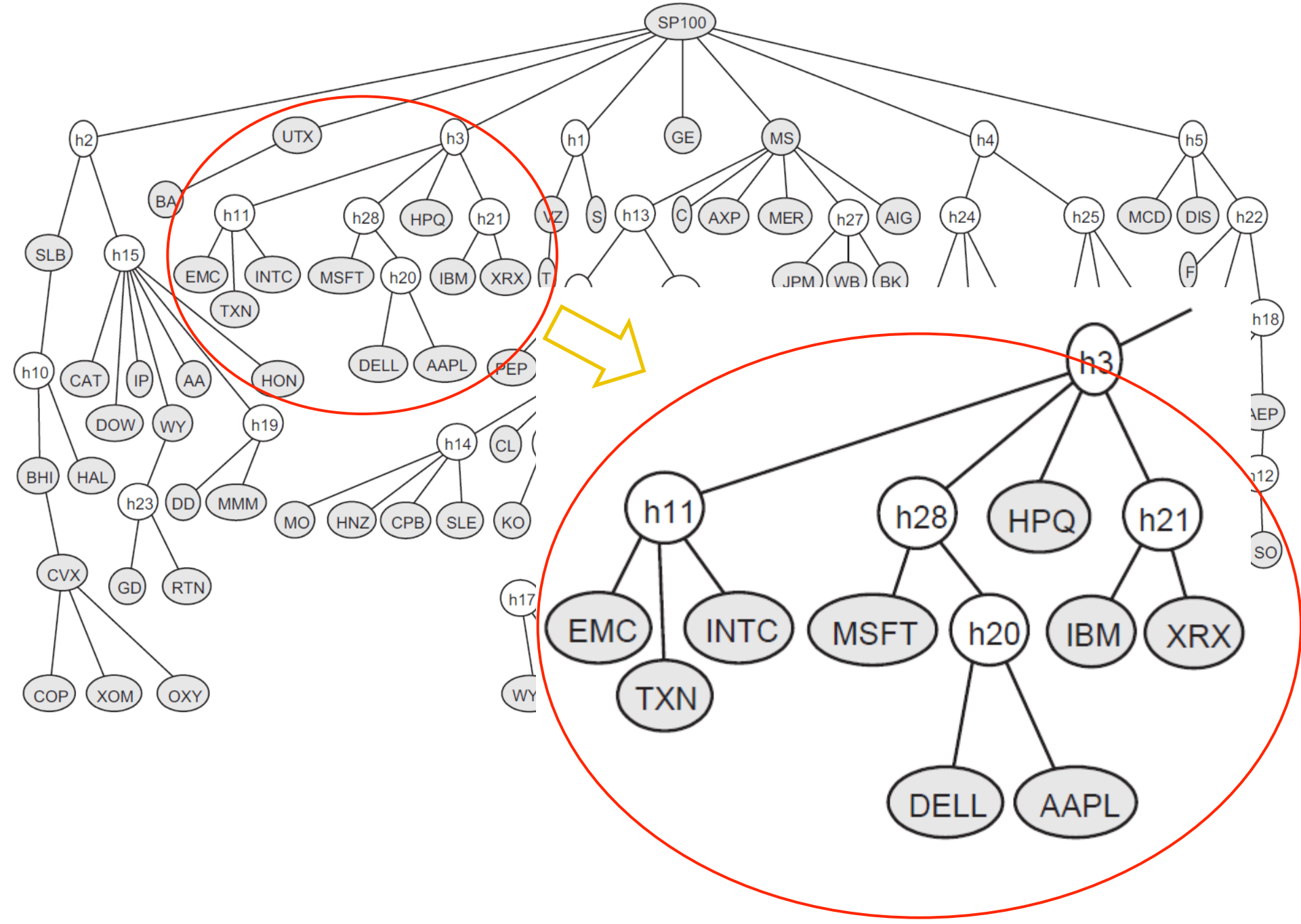
- Compute the ML estimates of information distances.
- Relaxed constraints for testing node relationships.
- Consistent (only in structure for discrete distributions)

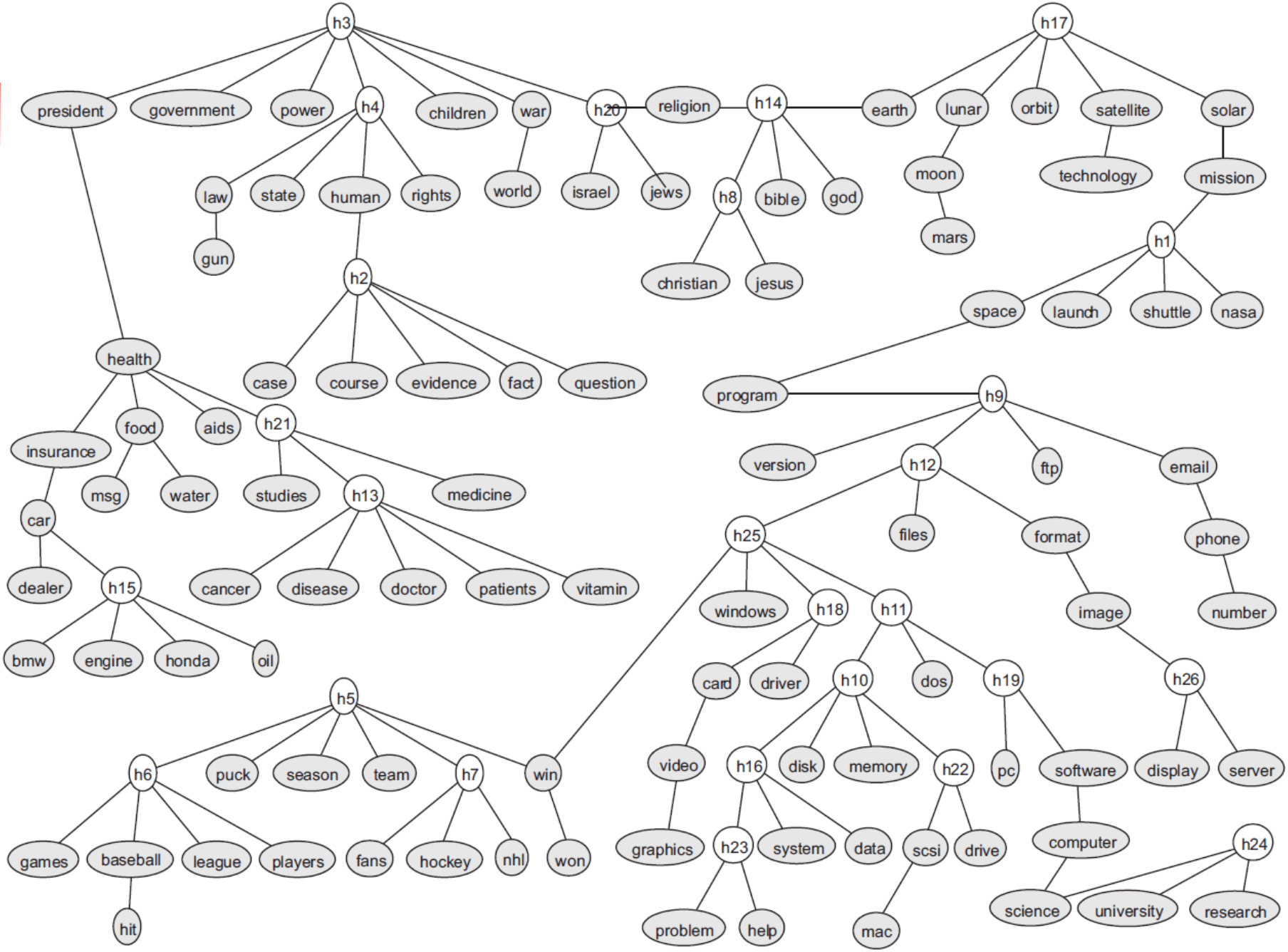
- Regularized CLGrouping for learning latent tree approximations.

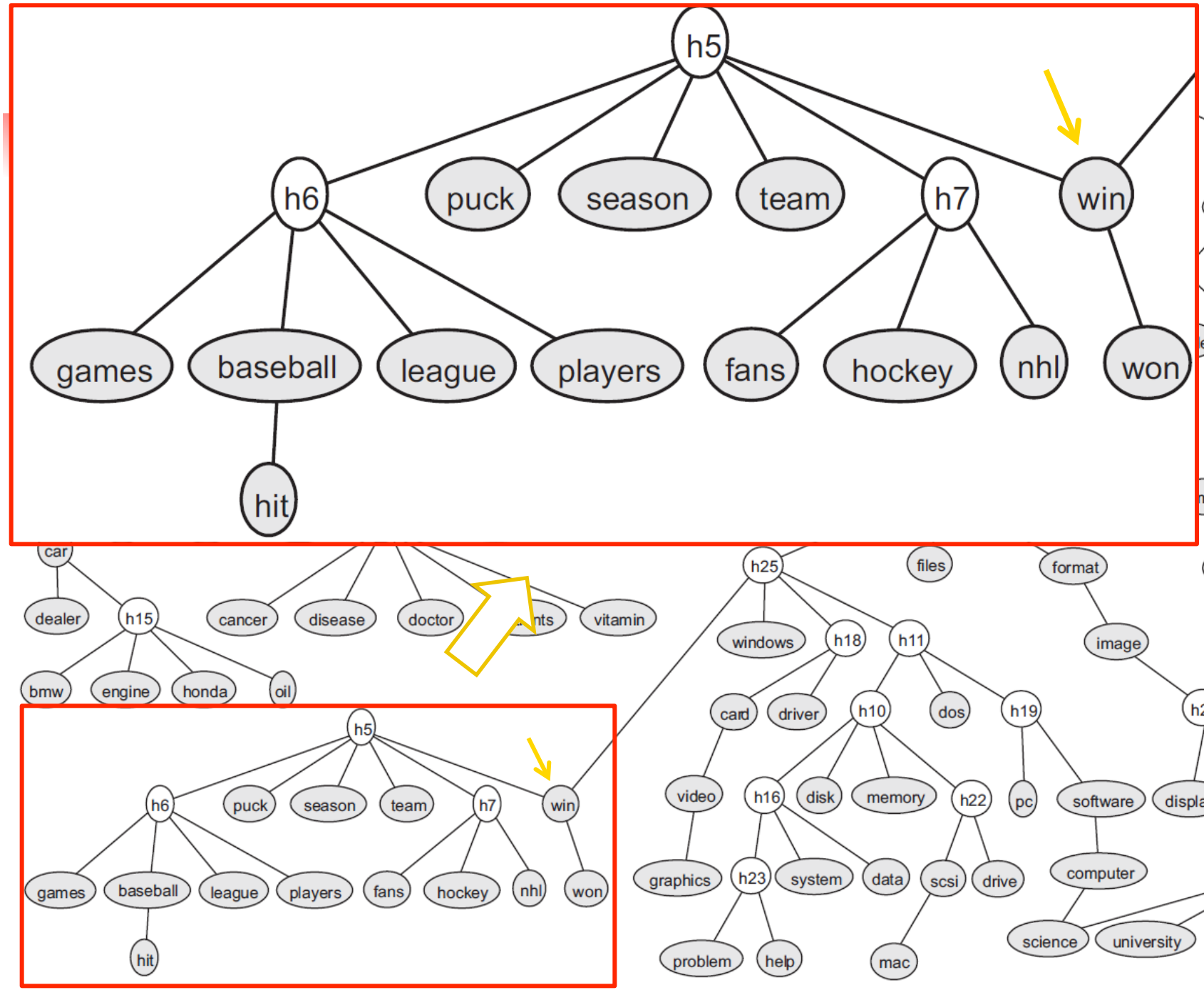

SP100
h2
UTX
h3
h1
GE
MS
h4
h5
BA
h11
h28
HPQ
h21
VZ
S
h13
C
AXP
MER
h27
AIG
h24
h25
MCD
DIS
h22
SLB
h15
EMC
INTC
MSFT
h20
IBM
XRX
T
h8
h29
JPM
WB
BK
F
CBS
ORCL
HD
WMT
TYC
h18
TXN
DELL
AAPL
PEP
h9
MDT
UNH
h26
SNS
CMCSA
TGT
h10
CAT
IP
AA
HON
DOW
WY
h19
FDX
CVS
WMB
AEP
h14
CL
h6
ABT
h7
AVP
BAC
RF
BNI
WFC
USB
BHI
HAL
h12
h23
DD
MMM
MO
HNZ
CPB
SLE
KO
PG
h16
AMGN
JNJ
NSC
EXC
ETR
SO
CVX
GD
RTN
h17
MRK
PFE
BMY
COP
XOM
OXY
WYE
BAX
CI

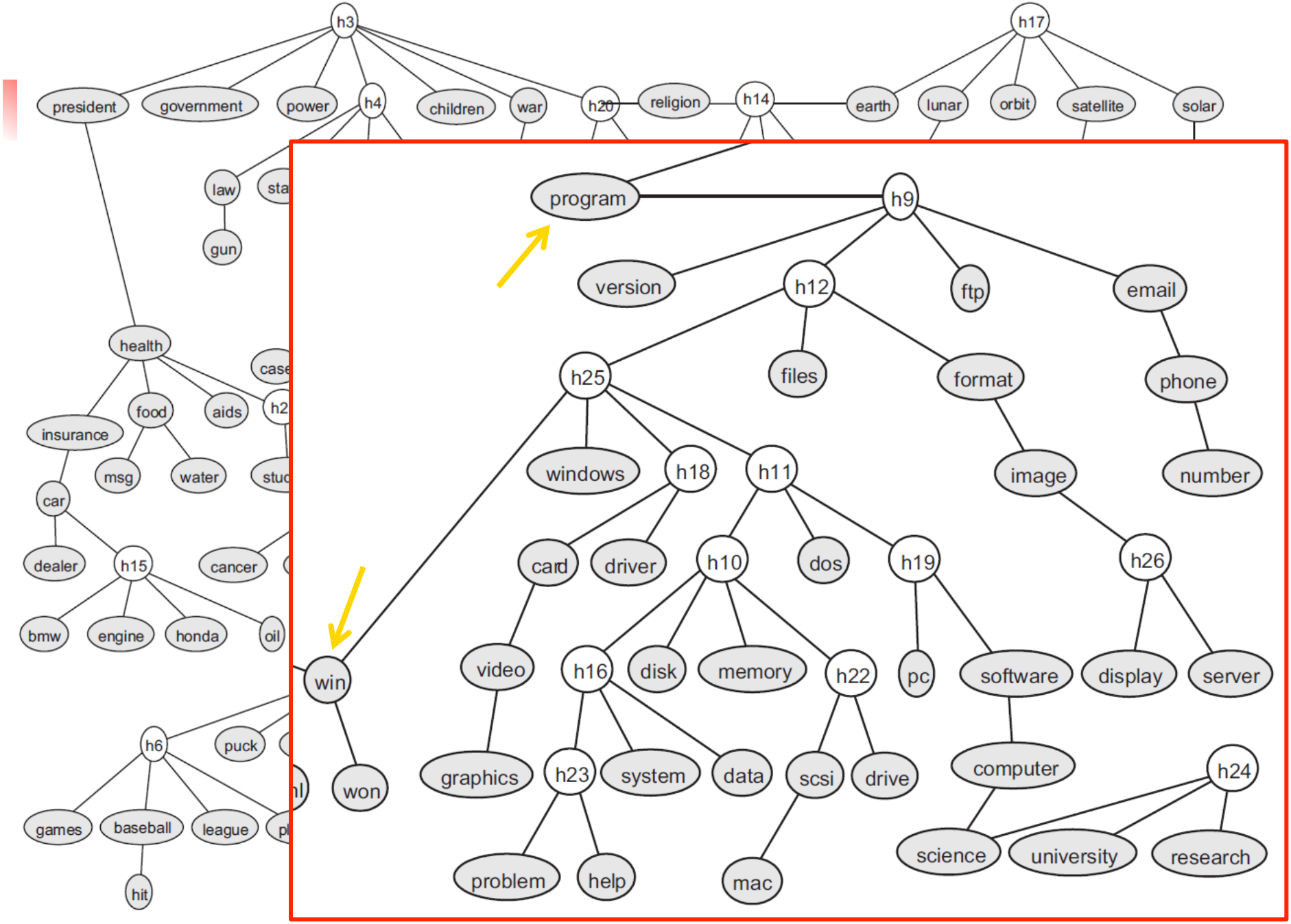

SP100
h2
UTX
h3
h1
GE
MS
h4
h5
BA
h11
h28
HPQ
h21
VZ
S
h13
C
AXP
MER
h27
AIG
h24
h25
MCD
DIS
h22
SLB
h15
EMC
INTC
MSFT
h20
IBM
XRX
T
h8
F
TXN
ORCL
HD
WMT
TYC
h18
h10
CAT
IP
AA
HON
DELL
AAPL
PEP
h9
MDT
UNH
TGT
FDX
CVS
WMB
AEP
DOW
WY
h19
h14
CL
h6
ABT
h7
AVP
BAC
BHI
HAL
h12
h23
DD
MMM
MO
HNZ
CPB
SLE
KO
PG
h16
AMGN
JNJ
EXC
ETR
SO
CVX
GD
RTN
h17
MRK
PFE
BMY
COP
XOM
OXY
WYE
BAX
CI
h1
VZ
S
T

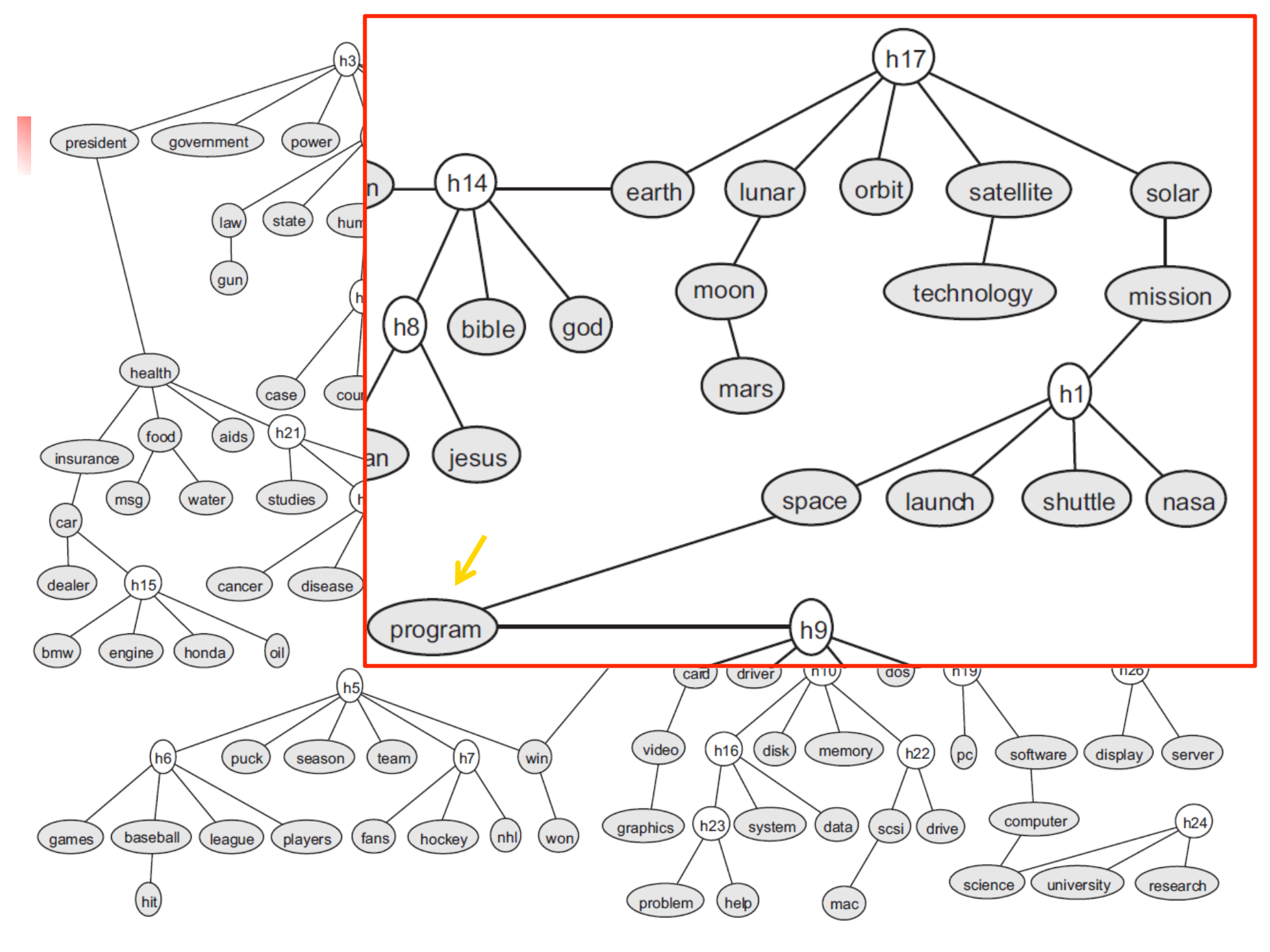

SP100
h2
UTX
h3
h1
GE
MS
h4
h5
BA
h11
h28
HPQ
h21
VZ
S
h13
C
AXP
MER
h27
AIG
h24
h25
MCD
DIS
h22
SLB
h15
EMC
INTC
MSFT
TXN
F
ORCL
HD
WMT
TYC
h18
h10
CAT
IP
AA
HON
DELL
CMCSA
TGT
DOW
WY
h19
FDX
CVS
WMB
AEP
BHI
HAL
USB
h12
h23
DD
MMM
MO
HNZ
EXC
ETR
SO
CVX
GD
RTN
COP
XOM
OXY
h25
HD
WMT
TGT
FDX
CVS

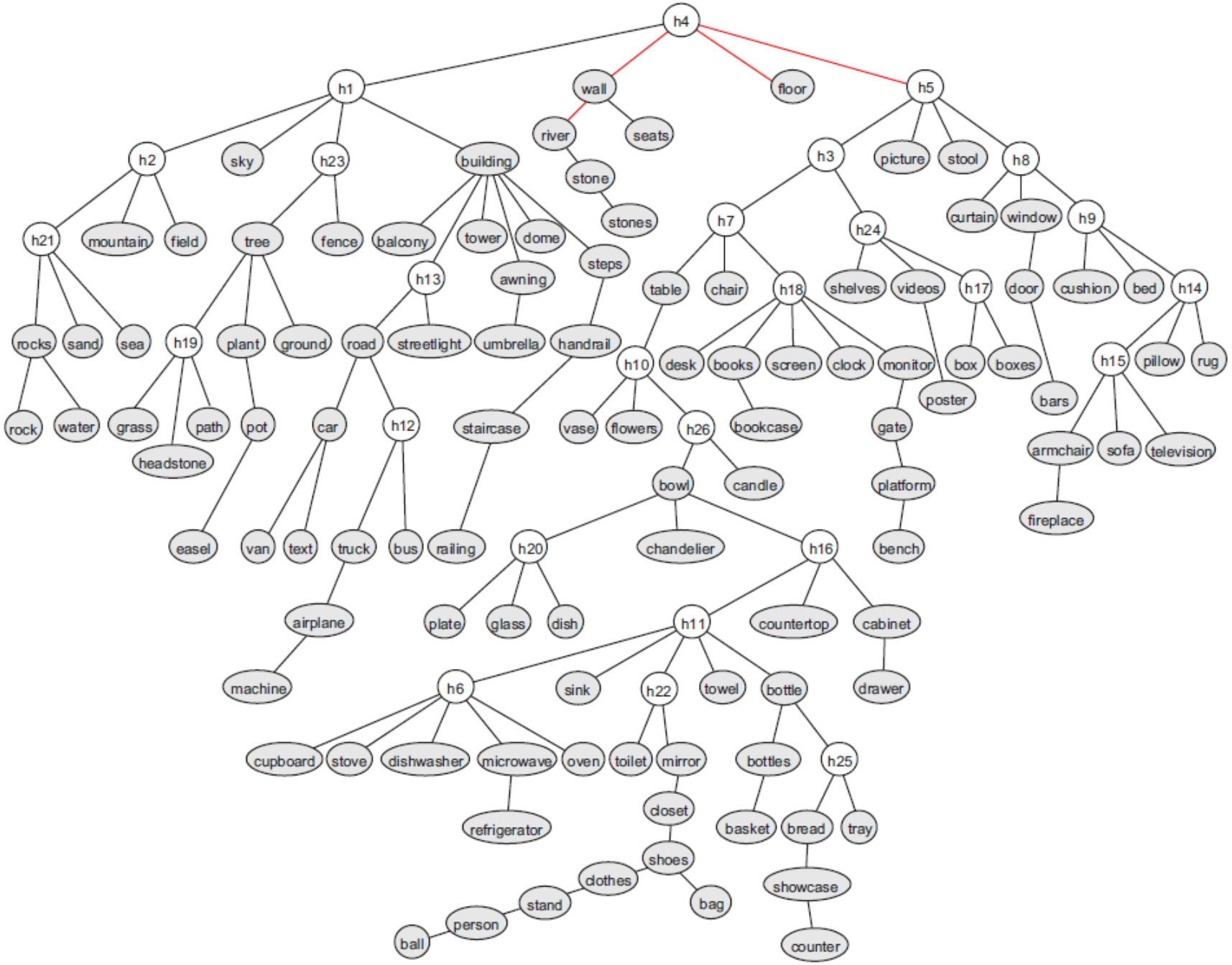

SP100
h2
UTX
h3
h1
GE
MS
h4
h5
BA
SLB
h15
h11
h28
HPQ
h21
VZ
S
h13
C
AXP
MER
h27
AIG
h24
h25
MCD
DIS
h22
F
EMC
INTC
MSFT
h20
IBM
XRX
T
JPM
WB
BK
TXN
h18
h10
CAT
IP
AA
HON
DELL
AAPL
PEP
DOW
WY
h19
h14
CL
BHI
HAL
h23
DD
MMM
MO
HNZ
CPB
SLE
KO
CVX
GD
RTN
h17
COP
XOM
OXY
WY
h3
h11
h28
HPQ
h21
EMC
INTC
MSFT
h20
IBM
XRX
TXN
DELL
AAPL

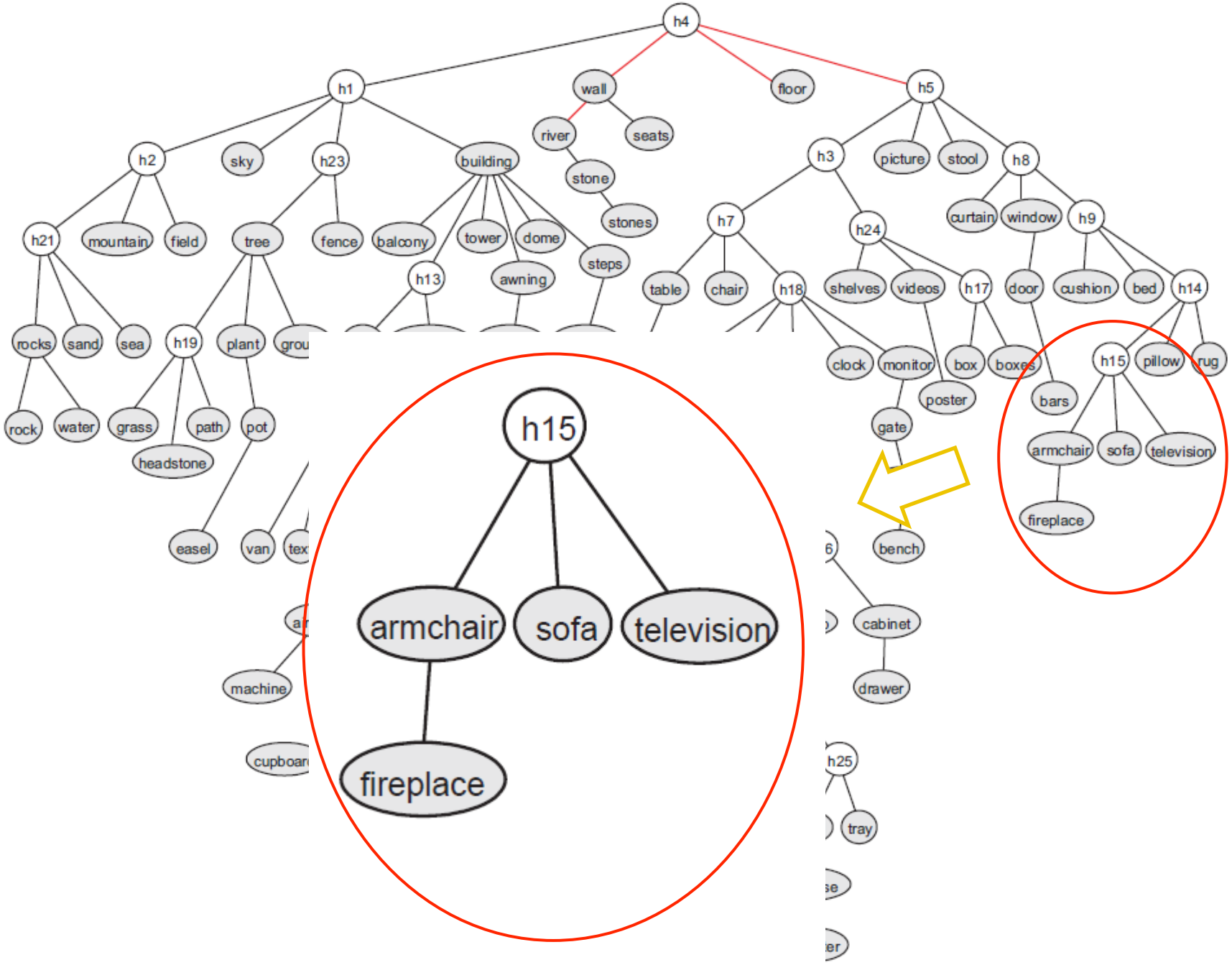

h3
president
government
power
h4
children
war
h20
religion
h14
earth
lunar
orbit
satellite
solar
h17
law
state
human
rights
world
israel
jews
h8
bible
god
moon
technology
mission
gun
mars
h1
h2
christian
jesus
space
launch
shuttle
nasa
health
case
course
evidence
fact
question
program
h9
insurance
food
aids
h21
version
h12
ftp
email
msg
water
studies
h13
medicine
h25
files
format
phone
car
dealer
h15
cancer
disease
doctor
patients
vitamin
windows
h18
h11
image
number
bmw
engine
honda
oil
card
driver
h10
dos
h19
h26
h5
video
h16
disk
memory
h22
pc
software
display
server
h6
puck
season
team
h7
win
graphics
h23
system
data
scsi
drive
computer
h24
games
baseball
league
players
fans
hockey
nhl
won
science
university
research
hit
problem
help
mac

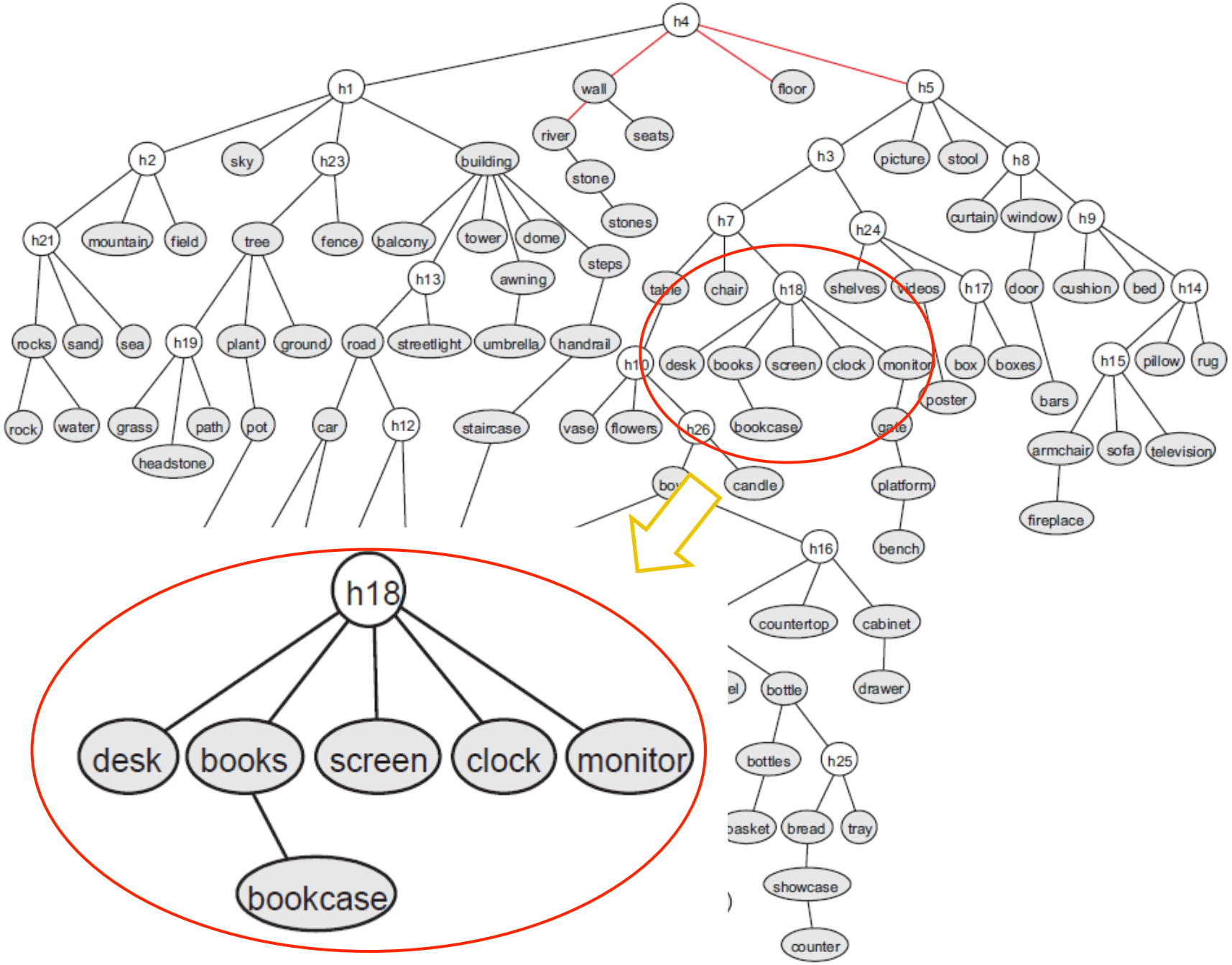

h5
h6
puck
season
team
h7
win
games
baseball
league
players
fans
hockey
nhl
won
hit
solar
mission
nasa
mail
phone
number
image
format
files
h25
windows
h18
h11
h10
dos
h19
h26
card
driver
video
h16
disk
memory
h22
pc
software
display
server
graphics
h23
system
data
scsi
drive
computer
h24
science
university
research
problem
help
mac
car
dealer
h15
cancer
disease
doctor
vitamin
bmw
engine
honda
oil


h3
president
government
power
h4
children
war
h20
religion
h14
h17
earth
lunar
orbit
satellite
solar
law
gun
program
h9
version
h12
ftp
email
health
h25
files
format
phone
food
aids
insurance
msg
water
windows
h18
h11
image
number
car
dealer
h15
cancer
card
driver
h10
dos
h19
h26
bmw
engine
honda
oil
win
video
h16
disk
memory
h22
pc
software
display
server
h6
puck
won
graphics
h23
system
data
scsi
drive
computer
h24
games
baseball
league
hit
problem
help
mac
science
university
research

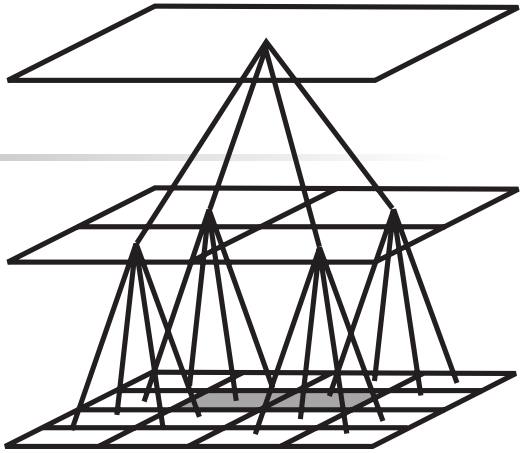

h3
president
government
power
law
state
gun
health
case
food
aids
h21
insurance
msg
water
studies
car
dealer
h15
cancer
disease
bmw
engine
honda
oil
h14
earth
lunar
orbit
satellite
solar
h8
bible
god
moon
technology
mission
mars
h1
jesus
space
launch
shuttle
nasa
h17
program
h9
h5
h6
puck
season
team
h7
win
games
baseball
league
players
fans
hockey
nhl
won
hit
video
h16
disk
memory
h22
graphics
h23
system
data
scsi
drive
problem
help
mac
pc
software
display
server
computer
h24
science
university
research


h4
h1
wall
floor
h5
river
seats
h2
sky
h23
building
h3
picture
stool
h8
stone
curtain
window
h9
stones
h7
h24
h21
mountain
field
tree
fence
balcony
tower
dome
steps
h13
awning
table
chair
h18
shelves
videos
h17
door
cushion
bed
h14
rocks
sand
sea
h19
plant
ground
road
streetlight
umbrella
handrail
h10
desk
books
screen
clock
monitor
box
boxes
h15
pillow
rug
poster
bars
rock
water
grass
path
pot
car
h12
staircase
vase
flowers
h26
bookcase
gate
armchair
sofa
television
headstone
bowl
candle
platform
fireplace
easel
van
text
truck
bus
railing
h20
chandelier
h16
bench
airplane
plate
glass
dish
h11
countertop
cabinet
machine
h6
sink
h22
towel
bottle
drawer
cupboard
stove
dishwasher
microwave
oven
toilet
mirror
bottles
h25
refrigerator
closet
basket
bread
tray
shoes
showcase
clothes
stand
bag
person
ball
counter


h4
h1
wall
floor
h5
river
seats
h2
sky
h23
building
h3
picture
stool
h8
stone
stones
curtain
window
h9
h21
mountain
field
tree
fence
balcony
tower
dome
h7
h24
steps
h13
awning
table
chair
h18
shelves
videos
h17
door
cushion
bed
h14
rocks
sand
sea
h19
plant
clock
monitor
box
boxes
h15
pillow
rug
poster
bars
rock
water
grass
path
pot
gate
armchair
sofa
television
headstone
fireplace
easel
van
bench
cabinet
machine
drawer
h25
tray
h15
armchair
sofa
television
fireplace


h4
h1
wall
floor
h5
river
seats
h2
sky
h23
building
h3
picture
stool
h8
stone
stones
curtain
window
h9
h21
mountain
field
tree
fence
balcony
tower
dome
h7
h24
steps
h13
awning
table
chair
h18
shelves
videos
h17
door
cushion
bed
h14
rocks
sand
sea
h19
plant
ground
road
streetlight
umbrella
handrail
h10
desk
books
screen
clock
monitor
box
boxes
h15
pillow
rug
poster
bars
rock
water
grass
path
pot
car
h12
staircase
vase
flowers
h26
bookcase
gate
armchair
sofa
television
headstone
candle
platform
fireplace
h16
bench
countertop
cabinet
bottle
drawer
bottles
h25
basket
bread
tray
showcase
counter
h18
desk
books
screen
clock
monitor
bookcase

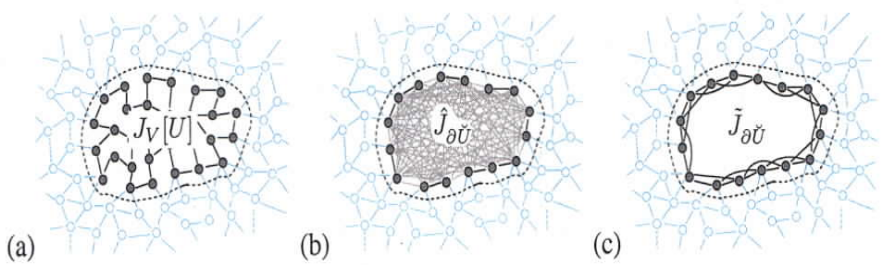
The dark side of trees = The bright side: No loops



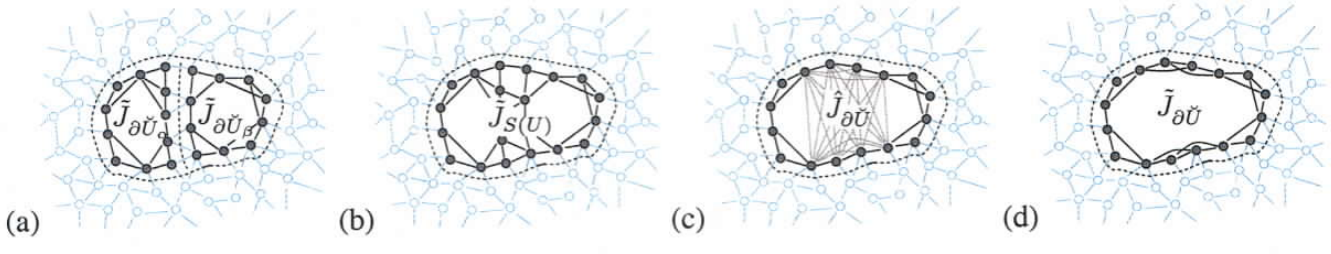
- So, what do we do?
- Try #1: Turn it into a junction tree
  - Not usually a good idea, but …
- Try #2: Pretend the problem isn't there and use a tree
  - If the real objectives are at coarse scales, then fine-scale artifacts may not matter
- Try #3: Pretend it's a tree and use (Loopy) BP
- Try #4: Think!
  - What does LBP ***do?***
  - Better algorithms?
  - Other graphs for which inference is scalable?

Recursive Cavity Models: "Reduced-order" modeling as part of estimation

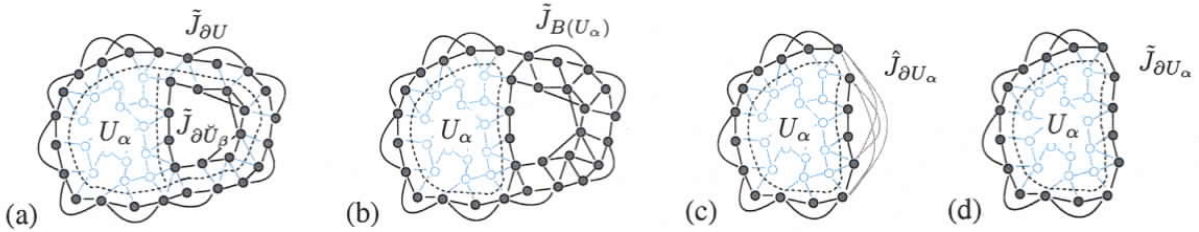
- Cavity thinning



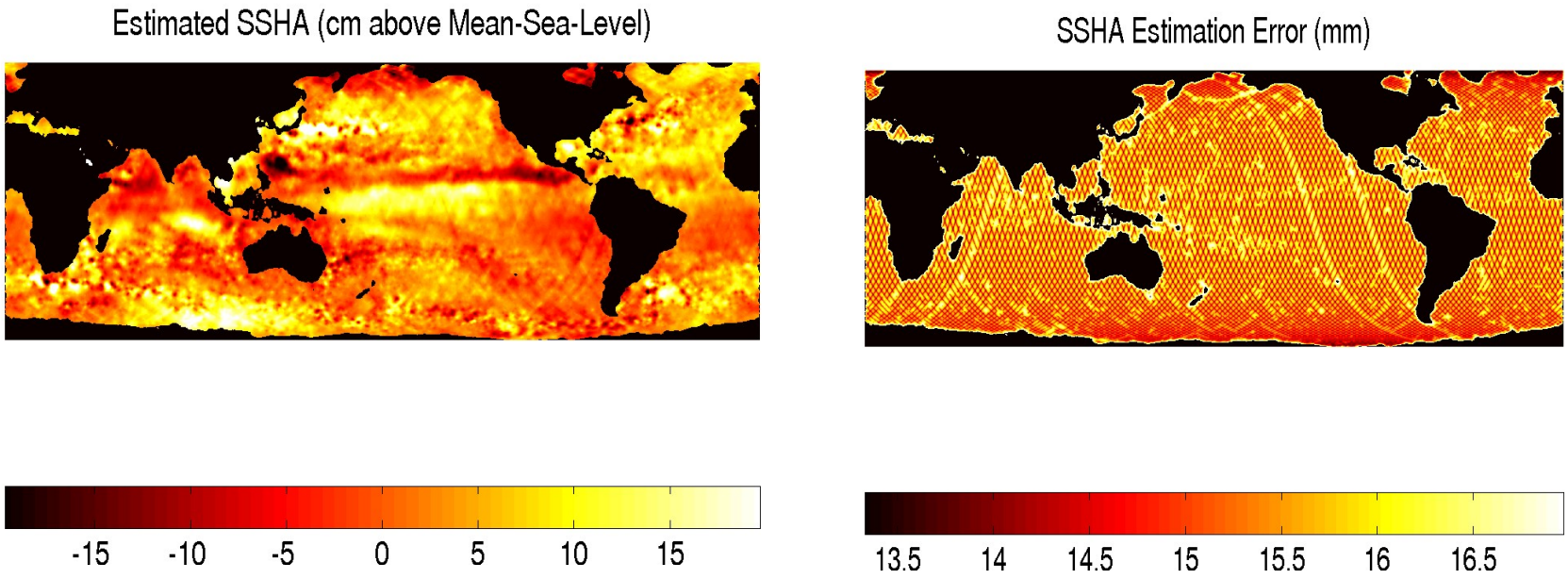
- Collision



- Reversing the process (bring your own blanket)



RCM in action: We are the world



- This is the information-form of RTS, with a thinning approximation at each stage
- How do the thinning errors propagate?
  A control-theoretic stability question

Walk-sums and Gaussian models

$$J = P^{-1} \qquad\qquad h = P^{-1}\mu$$

- Assume *J* normalized to have unit diagonal

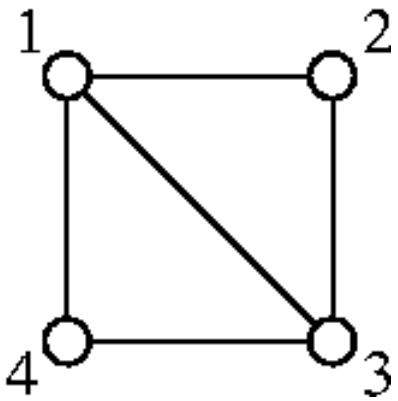
$$J^{-1} = (I - R)^{-1} = I + R + R^2 + \dots$$

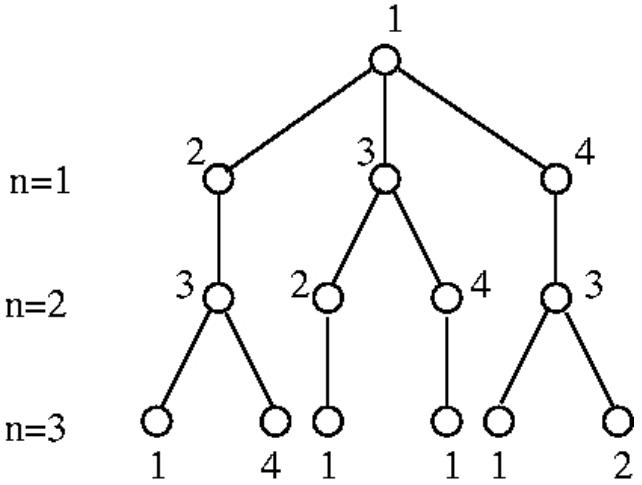
  - *R* is the matrix of ***partial correlation coefficients***
  - $(R^\ell)_{s,t}$ =***sum over weighted length-ℓ walks*** from *s* to *t* in graph

$$P_{s,t} = \phi(s \to t), \quad \mu_t = \sum_{s \in V} h_s \phi(s \to t)$$

- Inference algorithms may "collect" walks in different ways
- Walk-summability, corresponding to $\varrho(\bar{R}) < 1$, guarantees
  - Collection in any order is OK
  - LBP converges
  - If LBP converges it collects all walks for μ_i but only ***some*** of the ***self-return walks*** required for P_{ii}
  - There are lots of interesting/important models that are non-WS (and for which BP goes haywire)

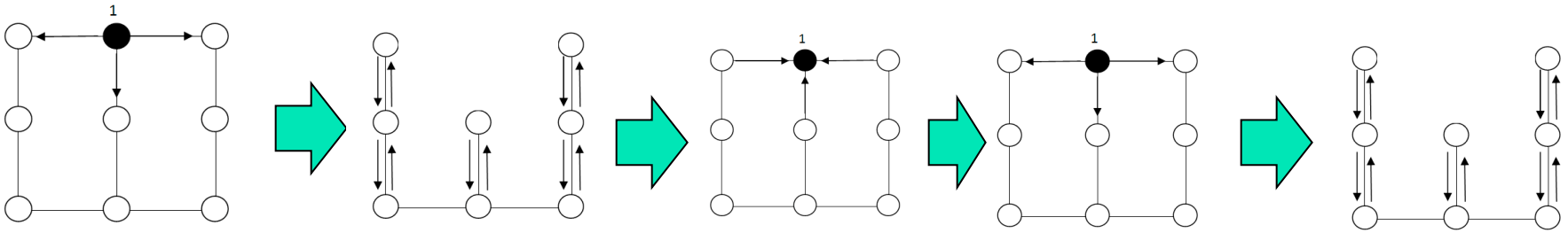
A computation tree





- BP includes the ***back-tracking self-return*** walk (1,2,3,2,1)
- BP does ***not*** include the walk (1,2,3,1)
- **BUT:** For Non-WS models, the tree **may be nonsensical**
- There are ways to collect some or all of the missed walks
  - Embedded subgraphs as preconditioners
    - Convergence for WS models always
  - A method that works also for non-WS models, recognizing that not all nodes are created equal

An alternate approach: Using (Pseudo-) ***Feedback Vertex Sets***

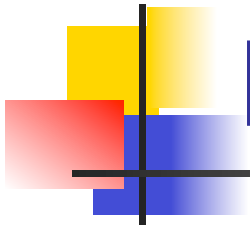


- Provide additional potentials to allow computation of quantities needed in mean/variance/covariance computation in the FVS
- Run BP with both original potentials and the additional set(s)
- Feed back information to FVS to allow computation of exact variance and mean within the FVS
- Send modified information potentials to neighbors of FVS
- Run BP with modified information potentials
  - Yields exact means immediately
  - Combining with results from Steps 2, 3 yields exact variances

Approximate FVS

- Complexity is $\mathcal{O}(k^2 n)$, where $k = |\mathcal{F}|$
- If k is too large
  - Use a pseudo- (i.e., partial) FVS, breaking only some loops
  - On the remaining graph, run LBP (or some other algorithm)
- Assuming convergence (which does ***not*** require WS)
  - Always get the correct means and variances on $\mathcal{F}$, exact means on $\mathcal{T}$, and (for LBP) approximate variances on $\mathcal{T}$
  - The approximate variances collect *more walks* than LBP on full graph
- Local (fast) method for choosing nodes for the pseudo-FVS to:
  - Enhance convergence
  - Collect the most important wants
- Theoretical and empirical evidence show $k \approx \mathcal{O}(\log n)$ works

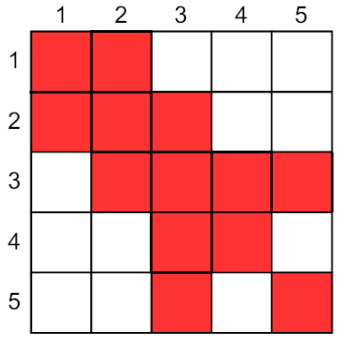
Motivation from PDEs: **MultiPOLE Models**

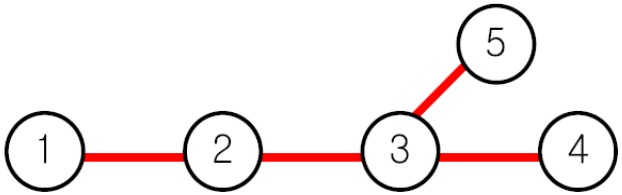


- Motivation from methods for efficient preconditioners for PDEs
  - Influence of variables at a distance are well-approximated by coarser approximation
  - We then only need to do ***LOCAL*** smoothing and correction
- The idea for statistical models:
  - Pyramidal structure in scale
  - However, when conditioned on neighboring scales, ***the remaining correlation structure at a given scale is sparse and local***

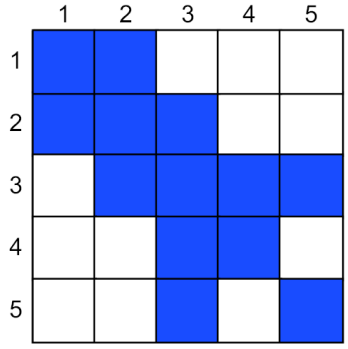
Models on graphs and on ***conjugate graphs***

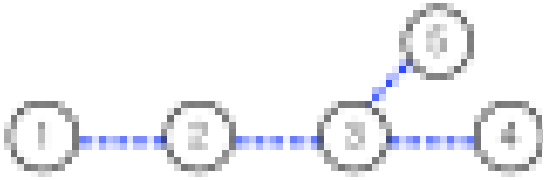
Garden variety graphical model: sparse ***inverse*** covariance

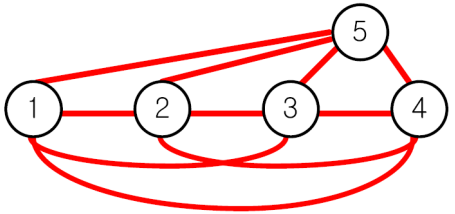




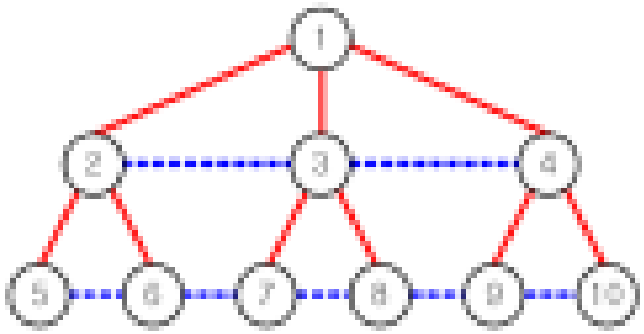
Conjugate models: sparse ***covariance***





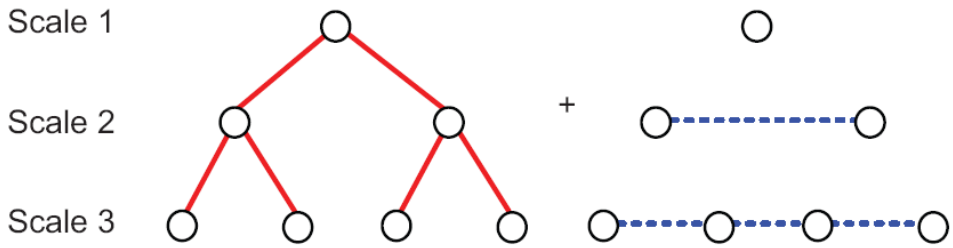


Inspiration from Multipole Methods for PDEs: Allow ***Sparse Residual Correlation*** Within Each Scale



- Conditioned on scale 1 and scale 3, x_2 is independent of x_4.

$$J = \begin{pmatrix} J_{[1]} & J_{[1,2]} & 0 \\ J_{[2,1]} & J_{[2]} & J_{[2,3]} \\ 0 & J_{[3,2]} & J_{[3]} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & J_{[1,2]} & 0 \\ J_{[2,1]} & 0 & J_{[2,3]} \\ 0 & J_{[3,2]} & 0 \end{pmatrix}}_{J^h} + \underbrace{\begin{pmatrix} J_{[1]} & 0 & 0 \\ 0 & J_{[2]} & 0 \\ 0 & 0 & J_{[3]} \end{pmatrix}}_{J^c = (\Sigma^c)^{-1}}$$

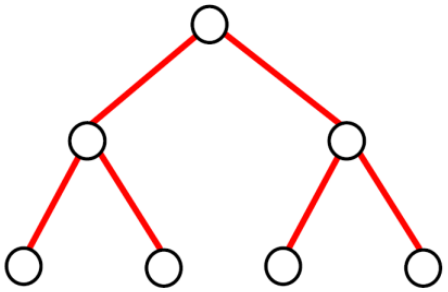


Learning such models:

"Dual" convex optimization problems

Multipole Estimation

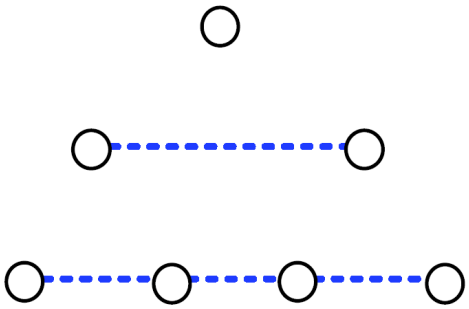
- Richardson Iteration to solve $(J^h + (\Sigma^c)^{-1})x = h$
  - Global tree-based inference
  - Sparse matrix multiplication for in-scale correction



$$J^h x_{new} = h - (\Sigma^c)^{-1} x_{old}$$

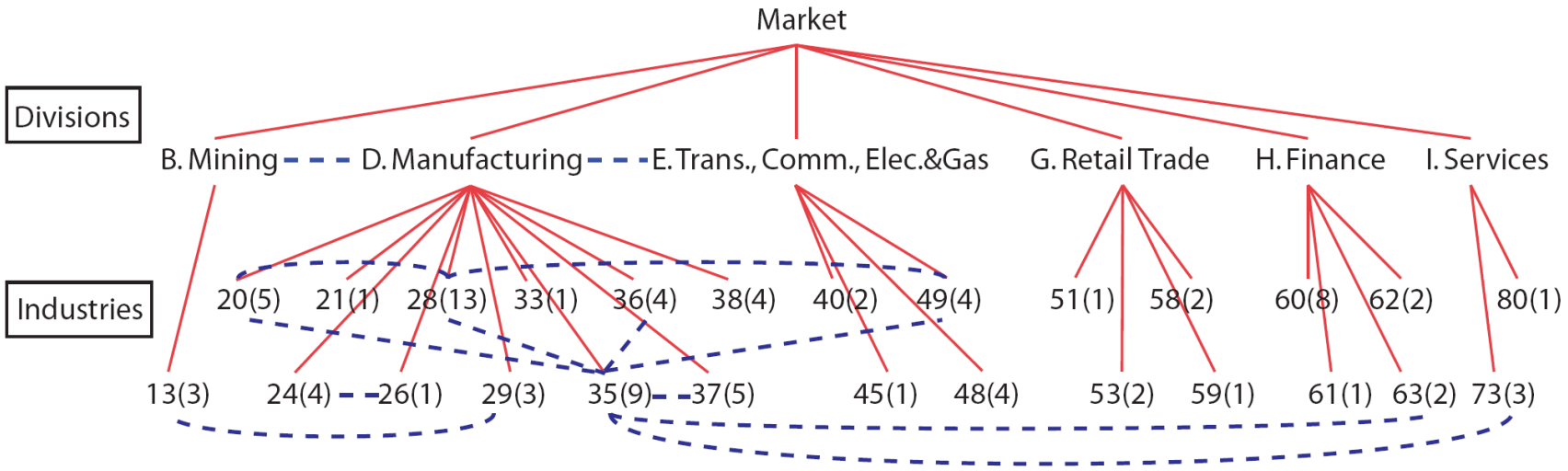
Compute last term via sparse equation

$$\Sigma^c z = x_{old}$$



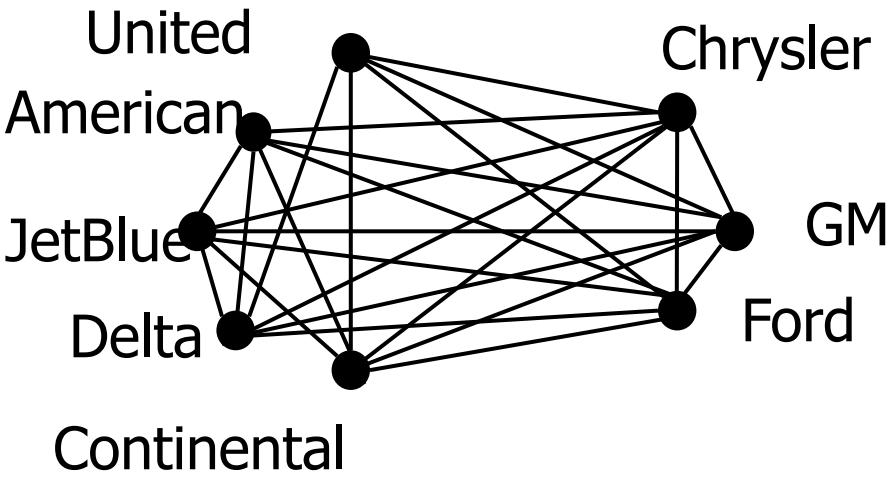
$$x_{new} = \Sigma^c (h - J^h x_{old})$$

Stock Returns Example

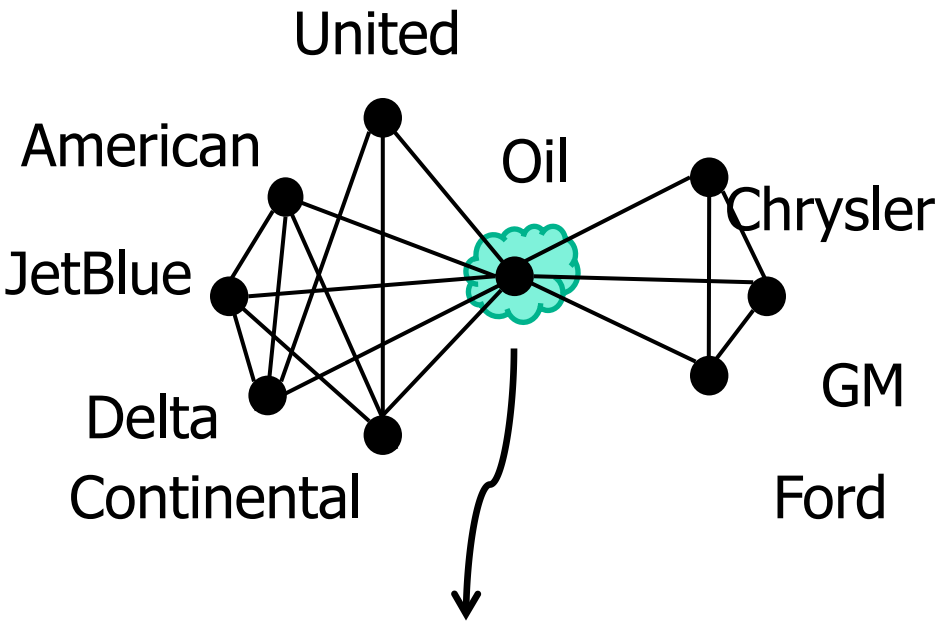


- Monthly returns of 84 companies in the S&P 100 index (1990-2007)
- Hierarchy based on the Standard Industrial Classification system
- Market, 6 divisions, 26 industries, and 84 individual companies
- Conjugate edges find strong residual correlations
  - Oil service companies (Schlumberger,…) and oil companies
  - Computer companies, Software companies, electrical equipment
  - …

What If Some Phenomena Hidden?



- **Many dependencies among observed variables**
- Less concise model



- **Hidden variables**
  - Hedge fund investments,
  - Patent accepted/rejected,
  - Geopolitical factors,
  - Regulatory issues, …

Graphical Models With Hidden Variables: Sparse Modeling Meets PCA

$$X = (X_O, X_H) \sim \mathcal{N}(0, \Sigma)$$

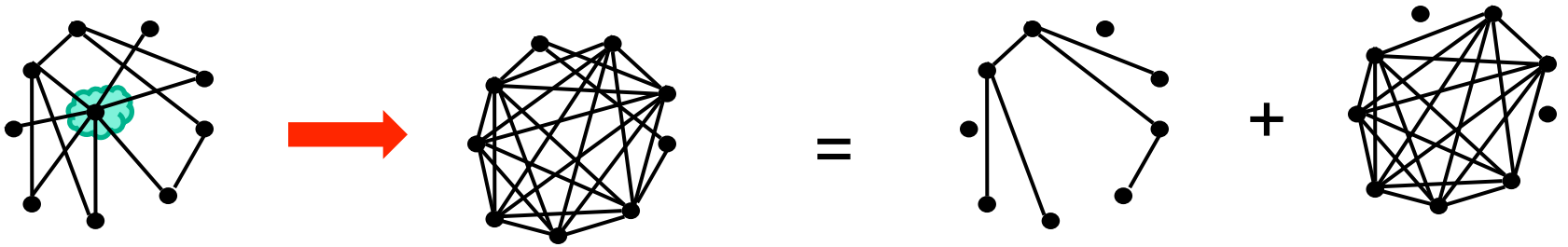
$$\Sigma = \begin{bmatrix} \Sigma_O & \Sigma_{O,H} \\ \Sigma_{H,O} & \Sigma_H \end{bmatrix} \qquad \Sigma^{-1} = K = \begin{bmatrix} K_O & K_{O,H} \\ K_{H,O} & K_H \end{bmatrix}$$

Marginal concentration matrix

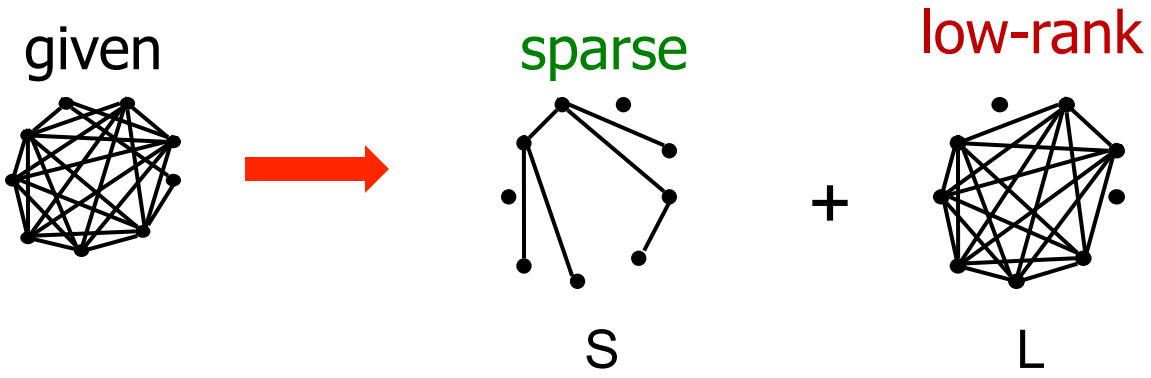
$$(\Sigma_O)^{-1} = K_O - K_{O,H} K_H^{-1} K_{H,O}$$

Sparse

Low-rank



Convex Optimization for Modeling



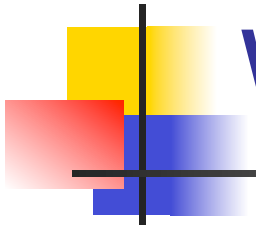
- Samples of obs. vars.: $\mathcal{D}_n = \{X_O^1, \ldots, X_O^n\}$

$$(\hat{S}_n, \hat{L}_n) = \arg\min_{S,L} \; -\tfrac{1}{n} \log \text{lik.}\,(S\text{–}L;\; \mathcal{D}_n) \;+\; \lambda_n \left[\gamma \|S\|_1 + \text{trace}(L)\right]$$

$$\text{s.t.} \quad S - L \succ 0, \;\; L \succeq 0.$$

- Last two terms provide convex regularization for sparsity in *S* and low-rank in *L*
- Weights allow tradeoff between sparsity and rank

When does this work?



- Identifiability conditions
  - The sparse part can't be low rank and the low rank part can't be sparse
- There are precise conditions (including conditions on regularization weights) that guarantee
  - Exact recovery if exact statistics are given
  - Consistency results if samples are available ("with high probability" scaling laws on problem dimension and available sample size)

On the way: Construct richer classes of models for which inference is ***easy***

- We have
  - Methods to learn hidden trees with fixed structure but unknown variable dimensions
  - Method to learn hidden trees with unknown structure but fixed (scalar) variable dimension
- Can we do both at the same time?
- We have method for recovering sparse plus low rank
  - How about ***tree*** plus low rank (i.e., small FVS)?
- Message-passing algorithms are distributed dynamic systems. How about designing better ones than LBP?