MPE, MAP AND APPROXIMATIONS

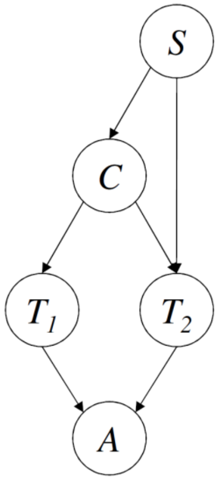
Statistical Methods in AI/ML

Vibhav Gogate
The University of Texas at Dallas

What we will cover?

- MPE= most probable explanation
  - The tuple with the highest probability in the joint distribution Pr(X|e)
- MAP=maximum a posteriori
  - Given a subset of variables Y, the tuple with the highest probability in the distribution P(Y|e)
- Exact Algorithms
  - Variable elimination
  - DFS search
  - Branch and Bound Search
- Approximations
  - Upper bounds
  - Local search

Running Example: Cheating in UTD CS Population



S	C	T_2	$\theta_{t_2 \mid c,s}$
male	yes	+ve	.80
male	yes	−ve	.20
male	no	+ve	.20
male	no	−ve	.80
female	yes	+ve	.95
female	yes	−ve	.05
female	no	+ve	.05
female	no	−ve	.95

T_1	T_2	A	$\theta_{a \mid t_1,t_2}$
+ve	+ve	yes	1
+ve	+ve	no	0
+ve	−ve	yes	0
+ve	−ve	no	1
−ve	+ve	yes	0
−ve	+ve	no	1
−ve	−ve	yes	1
−ve	−ve	no	0

S	θ_s
male	.55
female	.45

S	C	$\theta_{c \mid s}$
male	yes	.05
male	no	.95
female	yes	.01
female	no	.99

C	T_1	$\theta_{t_1 \mid c}$
yes	+ve	.80
yes	−ve	.20
no	+ve	.20
no	−ve	.80

Sex (S), Cheating (C), Tests (T1 and T2) and Agreement (A)

Most likely instantiations

- MPE = Most likely assignment to all non-evidence variables (given evidence)
- MAP = Most likely assignment to a subset of non-evidence variables (given evidence)
- A person takes a test and the test administrator says
  - The two tests agree (A = true)
- Query: Most likely instantiation of Sex and Cheating given evidence A = true
- Is this a MAP or an MPE problem?
- Answer: Sex=male and Cheating=no.

MPE vs MAP: Properties

- MPE is a special case of MAP
- Hardness
  - Computing MPE is NP-hard (Max-product problem)
  - Computing MAP is NP^{PP}-hard (Max-sum-product problem believed to be much harder than NP-hard)
- MPE projected on to the MAP variables does not yield the correct answer.
  - MPE given A=yes
    - S=female, C=no, T_1=negative and T_2=negative
  - MPE projected on MAP variables S and C
    - S=female, C=no is incorrect!
  - MAP given A=yes
    - S=male, C=no is correct!
- We will distinguish between
  - MPE and MAP probabilities
  - MPE and MAP instantiations

Bucket Elimination for MPE

- Same schematic algorithm as before
- Replace "elimination operator" by "maximization operator"

MAX_S

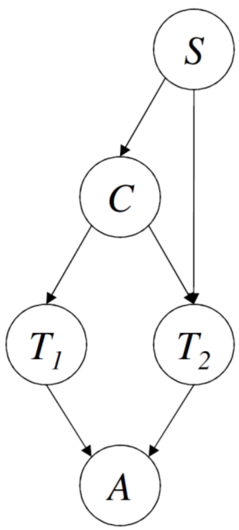
S	C	Value
male	yes	0.05
male	no	0.95
female	yes	0.01
female	no	0.99

=

C	Value
yes	0.05
no	0.99

Collect all instantiations that agree on all other variables except S and return the maximum value among them.

Bucket elimination: order (S, C, T1, T2)



$Factors: \phi(S)$

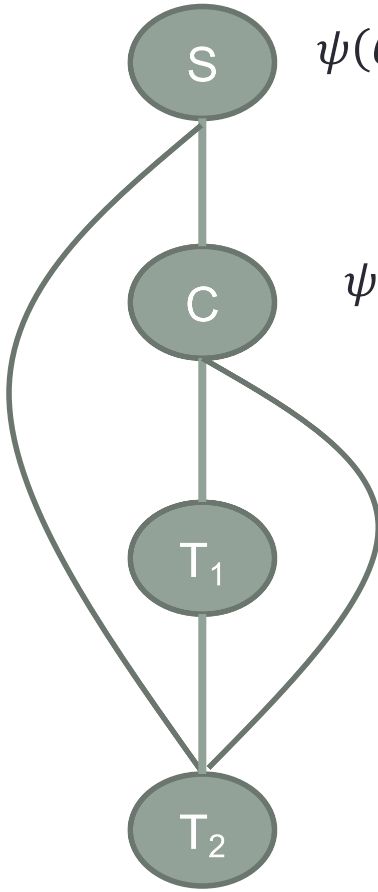
$\phi(C, S)$

$\phi(C, S, T_2)$

$\phi(C, T_1)$

$\phi(T_1, T_2)$

Evidence: A=true



$\psi(C, T_2)$ MAX_S $\phi(S)\phi(C, S)\phi(C, S, T_2)$

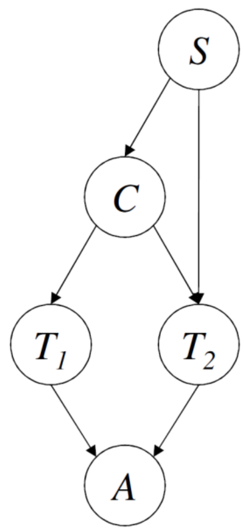
$\psi(T_1, T_2)$ MAX_C $\phi(C, T_1)$ $\psi(C, T_2)$

$\psi(T_2)$ MAX_{T1} $\phi(T_1, T_2)$ $\psi(T_1, T_2)$

MAX_{T2} $\psi(T_2)$

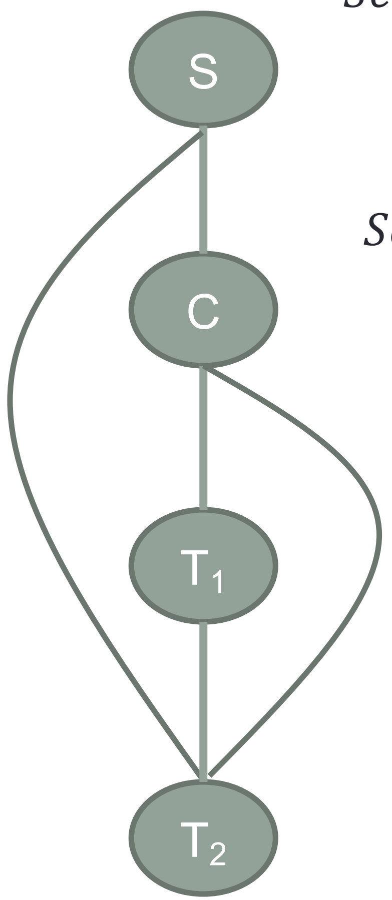
MPE probability

Bucket elimination: Recovering MPE tuple



$Factors\colon \phi(S)$
$\phi(C, S)$
$\phi(C, S, T_2)$
$\phi(C, T_1)$
$\phi(T_1, T_2)$

Evidence: A=true



$Set\ T_2 = -ve, C = no$
$\max(S\ tuple) = ?$
$S = female$

$\phi(S)\phi(C, S)\phi(C, S, T_2)$

$Set\ T_2 = -ve, T_1 = -ve$
$\max(C\ tuple) = ?$
$C = no$

$\phi(C, T_1)\ \ \psi(C, T_2)$

$Set\ T_2 = -ve$
$\max(T_1\ tuple) = ?$
$T_1 = -ve$

$\phi(T_1, T_2)\ \psi(T_1, T_2)$

$\max(T_2\ tuple) = ?$
$T_2 = -ve$

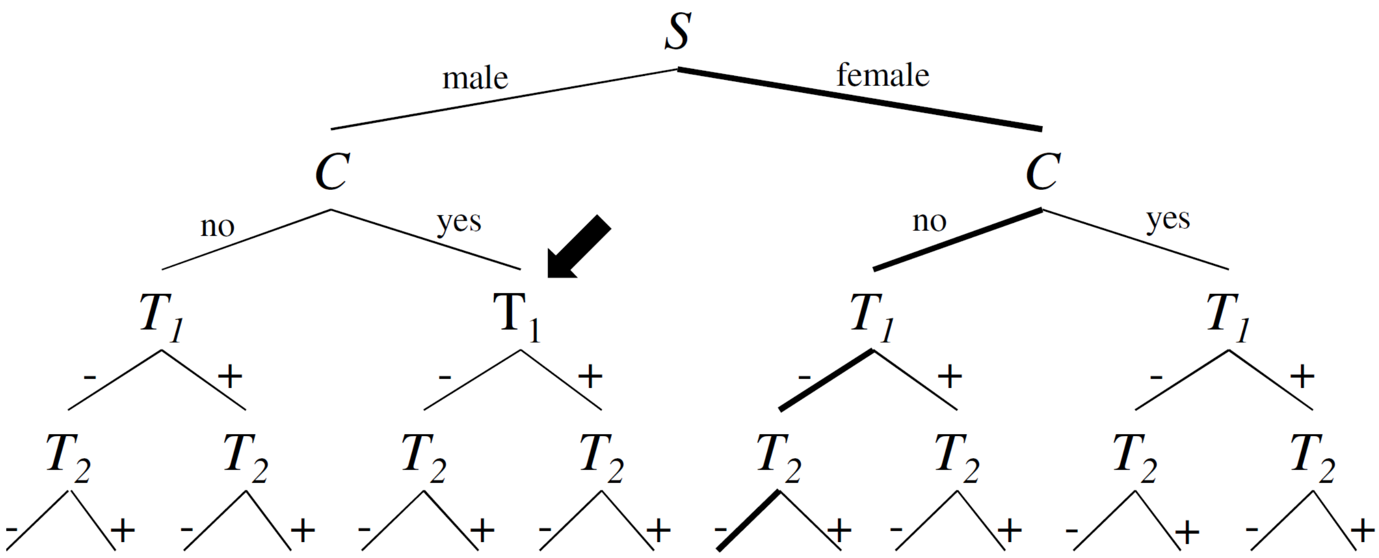
$\psi(T_2)$

MPE probability

Bucket elimination: MPE vs PE (Z)

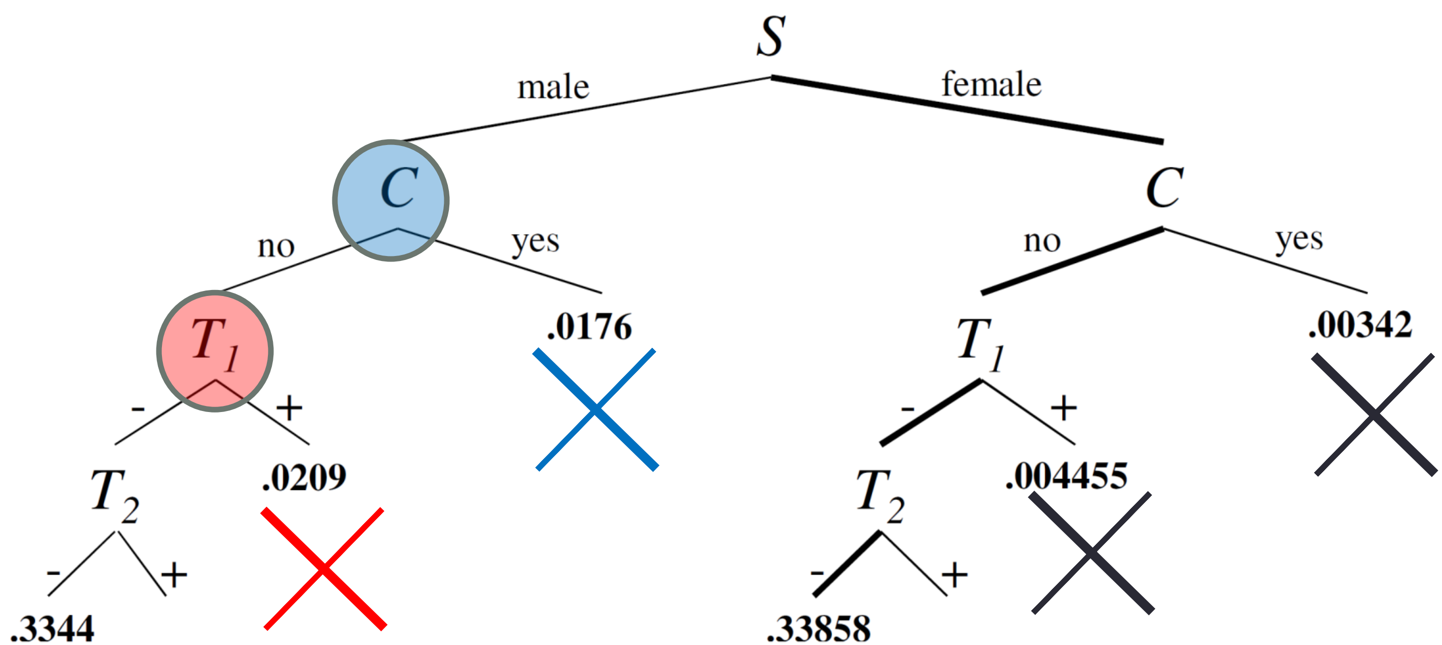
- Maximization vs summation
- Complexity: Same
  - Time and Space exponential in the width (w) of the given order: O(n exp(w+1)) timewise and O(n exp(w)) spacewise.

OR search for MPE



- At leaf nodes compute probabilities by taking product of factors
- Select the path with the highest leaf probability

Branch and Bound Search



- Let us say we have a method to upper bound MPE at each node
- Prune nodes which have smaller upper bound than the current MPE solution
- Amount of pruning depends on the quality of the upper bound. Lower the upper bound (i.e., better the upper bound), better the pruning.

Mini-Bucket Approximation: Idea

Split a bucket into mini-buckets => bound complexity

bucket (Y) =
$\{ \phi_1, \ldots, \phi_r, \phi_{r+1}, \ldots, \phi_n \}$

$$g = MAX_Y \left(\prod_{i=1}^{n} \phi_i \right)$$

$\{ \phi_1, \ldots, \phi_r \}$

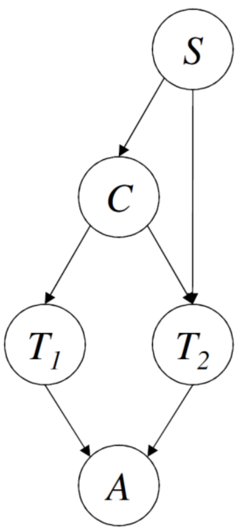
$$h_1 = MAX_Y \left(\prod_{i=1}^{r} \phi_i \right)$$

$\{ \phi_{r+1}, \ldots, \phi_n \}$

$$h_2 = MAX_Y \left(\prod_{i=r+1}^{n} \phi_i \right)$$

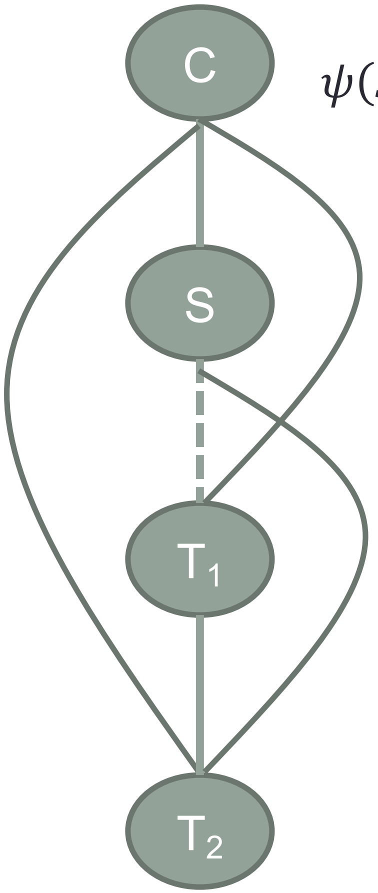
$$g \leq h_1 \times h_2$$

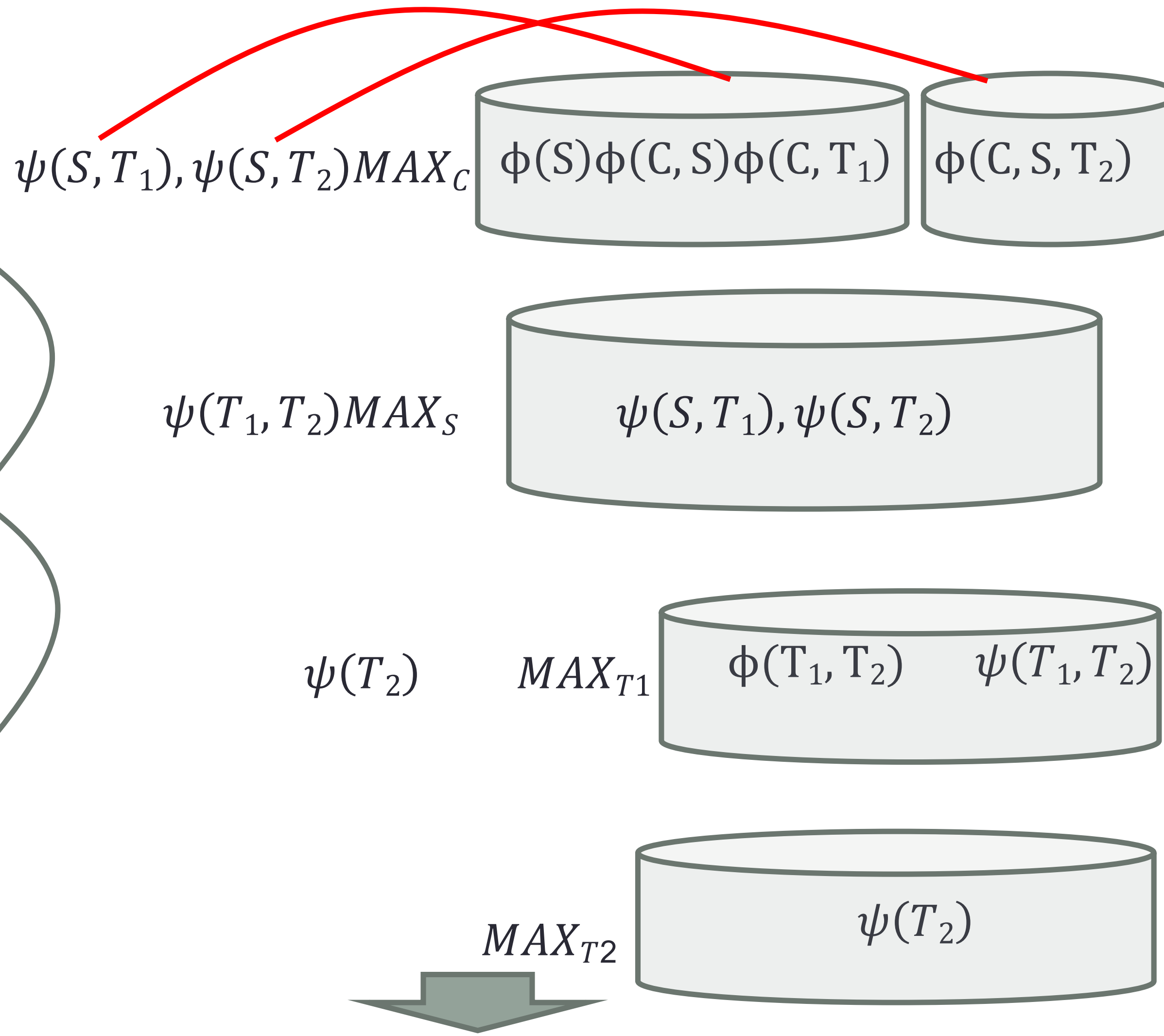
Mini Bucket elimination: (max-size=3 vars)



$Factors: \phi(S)$
$\phi(C,S)$
$\phi(C,S,T_2)$
$\phi(C,T_1)$
$\phi(T_1,T_2)$

Evidence: A=true

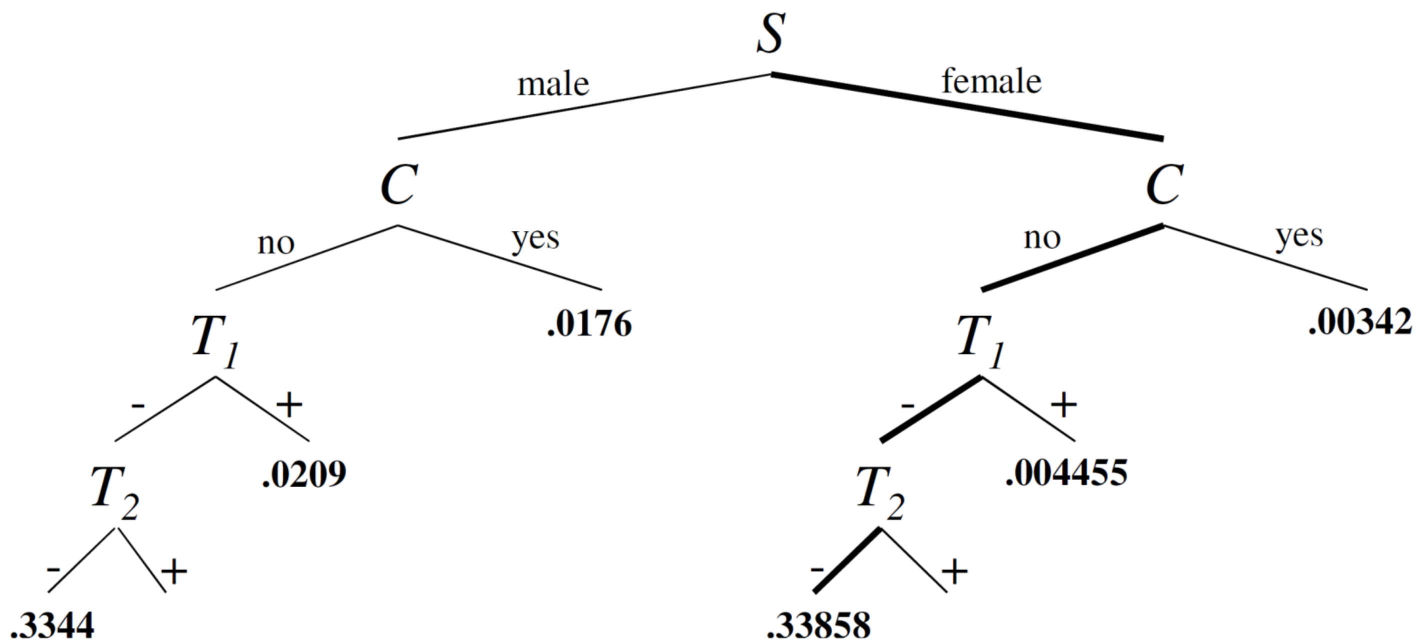




Mini-bucket (i-bounds)

- A parameter "i" which controls the size of (number of variables in) each mini-bucket
- Algorithm exponential in "i" : O(n exp(i))
- Example
  - i=2, quadratic
  - i=3, cubed
  - etc
- Higher the i-bound, better the upper bound
- In practice, can use i-bounds as high as 22-25.

Branch and Bound Search

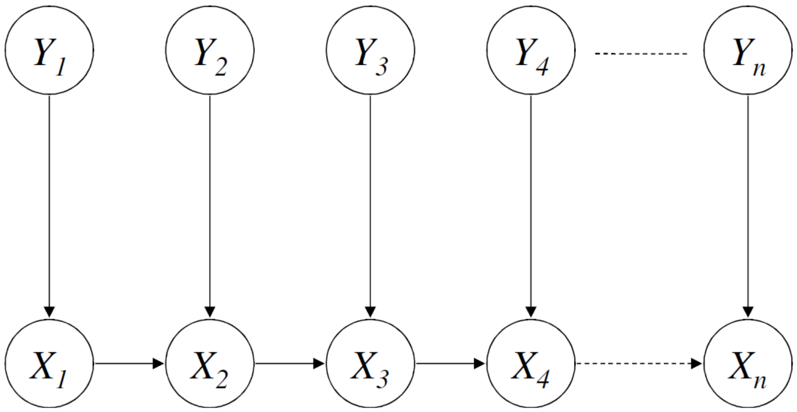


- Run MBE at each branch point.
- Prune nodes which have smaller upper bound than the current MPE solution
- Radu Marinescu's PhD thesis (AND/OR branch and bound search plus more): https://www.ics.uci.edu/~dechter/publications/r158.pdf

Computing MAP probabilities: Bucket Elimination

- Given MAP variables “M” and evidence be “e”
- Can compute the MAP probability using bucket elimination by first summing out all non-MAP variables, and then maximizing out MAP variables.
- By summing out non-MAP variables we are effectively computing the joint marginal Pr(M, e) in factored form.
- By maximizing out MAP variables M, we are effectively solving an MPE problem over the resulting marginal.
- The variable order used in BE_MAP is constrained as it requires MAP variables M to appear last in the order.
- Best case: BE_MAP is exponential in constrained treewidth which is the minimum width over (constrained) orders in which non-MAP variables are ordered before MAP variables.

MAP and constrained width



- Treewidth = 2
- MAP variables = $\{Y_1,..,Y_n\}$
- Any order in which M variables come first has width greater than or equal to n
- BE_MPE is exponential in 3 and BE_MAP is exponential in O(n).

MAP by branch and bound search

- MAP can be solved using depth-first brand-and-bound search, just as we did for MPE.
- Algorithm BB_MAP resembles the one for computing MPE with two exceptions.
- Exception 1: The search space consists only of the MAP variables
- Exception 2: We use a version of MBE_MAP for computing the bounds
  - Order all MAP variables after the non-MAP variables.

MAP by Local Search

- Given a network with n variables and an elimination order of width w
  - Complexity: O(r nexp(w+1)) where “r” is the number of local search steps
- Start with an initial random instantiation of MAP variables
- Neighbors of the instantiation “m” are instantiations that result from changing the value of one variable in “m”
- Score for neighbor “m”: Pr(m,e)
- How to compute Pr(m,e)?
  - Bucket elimination.

MAP: Local search algorithm

LS_MAP($\mathcal{N}$, **M**, **e**)

input:

$\mathcal{N}$: Bayesian network
M: some network variables
e: evidence ($\mathbf{E} \cap \mathbf{M} = \emptyset$)

output: instantiation **m** of **M** which (approximately) maximizes $\Pr(\mathbf{m}|\mathbf{e})$.

main:

$r \leftarrow$ number of local search steps
$P_f \leftarrow$ probability of randomly choosing a neighbor
$\mathbf{m}^\star \leftarrow$ some instantiation of variables **M** {best instantiation}
$\mathbf{m} \leftarrow \mathbf{m}^\star$ {current instantiation}
for r times **do**
  $p \leftarrow$ random number in [0, 1]
  if $p \leq P_f$ **then**
    $\mathbf{m} \leftarrow$ randomly selected neighbor of **m**
  else
    compute the score $\Pr(\mathbf{m} - X, x, \mathbf{e})$ for each neighbor $\mathbf{m} - X, x$
    if no neighbor has a higher score than the score for **m** **then**
      $\mathbf{m} \leftarrow$ randomly selected neighbor of **m**
    else
      $\mathbf{m} \leftarrow$ a neighbor of **m** with a highest score
    end if
  end if
  if $\Pr(\mathbf{m}, \mathbf{e}) > \Pr(\mathbf{m}^\star, \mathbf{e})$, **then** $\mathbf{m}^\star \leftarrow \mathbf{m}$
end for
return $\mathbf{m}^\star$

Recap

- Exact MPE and MAP
  - Bucket elimination
  - Branch and Bound Search
- Approximations
  - Mini bucket elimination
  - Branch and Bound Search
  - Local Search