Probabilistic Models: Spring 2014
Structure Learning with Dynamic Programming
Example Solutions

We are given the following local scores for some decomposable scoring function and dataset $\mathcal{D}$.

$Score(A, PA_A : \mathcal{D})$	PA_A
44.10	C E
43.55	B E
46.90	E
47.93	D
45.66	B
48.17	$\emptyset$

$Score(B, PA_B : \mathcal{D})$	PA_B
15.11	C E
8.40	C D
30.90	D
25.37	C
31.08	A
24.11	A C
33.59	$\emptyset$

$Score(C, PA_C : \mathcal{D})$	PA_C
8.40	B E
25.23	A E
8.40	B D
28.04	E
30.90	D
40.25	B
48.47	$\emptyset$

$Score(D, PA_D : \mathcal{D})$	PA_D
8.40	B E
8.40	B C
10.90	E
25.37	C
40.25	B
42.70	A
24.11	A C
42.94	$\emptyset$

$Score(E, PA_E : \mathcal{D})$	PA_E
13.81	B C
12.47	D
24.07	C
43.24	A
17.95	A C
44.51	$\emptyset$

1. Calculate the score of the optimal network according to these scores. Assume we want to minize the network score.

   - We find the scores of the singleton subnetworks (e.g., $\mathbf{U} = \{A\}$) by trivially selecting the empty parent sets. For example, we see that $Score(\{A\}) = 48.17$ and $Score(\{B\}) = 33.59$.
   - We calculate the scores of the two-variable subnetworks by adding each possible leaf to each of the singleton subnetworks according to either the algorithm or the recurrence. For example, we can calculate $Score(\{A, B\})$ using the recurrence as follows.

$$Score(\{A, B\}) = \min \left\{ \begin{array}{c} \{Score(\{A\}) + BestScore(B, \{A\})\}, \\ \{Score(\{B\}) + BestScore(A, \{B\})\} \end{array} \right\}$$

   We can calculate $BestScore(A, \{B\})$ by simply scanning through the list of scores and looking for the smallest value for parent sets which are subsets of $\{B\}$. In this case, $BestScore(A, \{B\}) = 45.66$, which happens when we select the parents of A to be $\{B\}$.

We can calculate the remaining scores for two-variable subnetworks similarly. The calculations for the three-variable subnetworks continues analogously. For example, $Score(\{A, B, C\})$ is calculated using the recurrence as follows.

$$Score(\{A,B,C\}) = \min \left\{ \begin{array}{r} \{Score(\{A,B\}) + BestScore(C, \{A,B\})\}, \\ \{Score(\{A,C\}) + BestScore(B, \{A,C\})\}, \\ \{Score(\{B,C\}) + BestScore(A, \{B,C\})\} \end{array} \right\}$$

We repeat this process layer-by-layer until we have the score for all subnetworks. The scores $Score(\mathbf{U})$ is given for all subnetworks below.

$\mathbf{U}$	$Score(\mathbf{U})$
$\emptyset$	0.00
{A}	48.18
{B}	33.59
{C}	48.48
{D}	42.95
{E}	44.51
{A, B}	79.26
{A, C}	96.65
{B, C}	73.85
{A, D}	90.88
{B, D}	73.85
{C, D}	73.85
{A, E}	91.42
{B, E}	78.11
{C, E}	72.56
{D, E}	55.42

$\mathbf{U}$	$Score(\mathbf{U})$
{A, B, C}	119.51
{A, B, D}	119.51
{A, C, D}	120.77
{B, C, D}	82.26
{A, B, E}	121.67
{A, C, E}	114.61
{B, C, E}	86.52
{A, D, E}	102.33
{B, D, E}	86.33
{C, D, E}	83.46
{A, B, C, D}	127.92
{A, B, C, E}	129.72
{A, B, D, E}	129.88
{A, C, D, E}	125.52
{B, C, D, E}	91.87
{A, B, C, D, E}	133.93

Useful Algorithms

Notation

- $Score(\mathbf{U})$. The score of the optimal subnetwork over variables $\mathbf{U}$
- $BestScore(X, \mathbf{U})$. The score of the best parent set for X which is a subset of $\mathbf{U}$
- $|\mathbf{U}|$. The number of variables in $\mathbf{U}$

```
procedure EXPAND(node U, sorted family scores BestScore)
    for each leaf in V \ U do
        newScore ← Score(U) + BestScore(leaf, U)
        if newScore < Score(U ∪ leaf) then
            Score(U ∪ leaf) ← newScore
        end if
    end for
end procedure

procedure MAIN(variables V, sorted family scores BestScore)
    Score(∅) ← 0
    for layer l = 0 to |V| do
        for each node U such that |U| = l do
            expand(U, BestScore)
        end for
    end for
    return Score(V)
end procedure
```