Motivation

- Many inference methods require a tree decomposition (TD)
  - Complexity is exponential in TD width
- Finding minimal width TD is NP-hard
- Leads to difficult dilemma:
  - How much time should I spend searching for a better TD?
- In practice, just run a heuristic for $k$ iterations
- Desire a more principled approach
Illustration

- Should we stop at points A, B or some future time C?
Formulation

- Let $C(w)$ be total cost of answering query
  
  $$C(w) = C_{\text{srch}}(w) + C_{\text{comp}}(w)$$

- Example:
  
  - Let $w_1$ and $w_2$ be two induced widths $w_1 < w_2$
  - Assume $C_{\text{srch}}(w_1) > C_{\text{srch}}(w_2)$ & $C_{\text{comp}}(w_1) < C_{\text{comp}}(w_2)$
  - If have taken $C_{\text{srch}}(w_2)$ to find ordering, continue if
    
    $$C_{\text{srch}}(w_1) + C_{\text{comp}}(w_1) < C_{\text{srch}}(w_2) + C_{\text{comp}}(w_2)$$

- Want to find ordering that minimizes $C(w)$
Formulation...

- Don’t know $C_{\text{srch}}(w)$ and $C_{\text{comp}}(w)$
- $C_{\text{srch}}(w)$ is a random quantity
  - Depends on stochastic heuristic (e.g. min-fill) and problem instance
- $C_{\text{comp}}(w)$ is function of problem instance and elimination order
  - Approximate by instance features and width
Estimating $C_{srch}()$

- Assume we run a heuristic repeatedly.
- After each iteration we record:
  1. The induced width of the ordering found, $X_i$
  2. The time, $T_i$
- Model iterations as events in a Poisson Process with rate $\lambda$
- We classify each event as one of $m$ widths.
- If independent events, have $m$ independent Poisson Processes with rates $\lambda_j = \lambda p_j$.
Estimating $C_{srch}()$

- Estimate $\lambda$ and $p_j$ ($j=1...m$) given observations $X_1...X_N$ and $T_1...T_N$
- Posterior is

$$f(p_1,..., p_m, \lambda | X^{(i)}) \propto (\lambda T_i)^{N_i} \exp \{-\lambda T_i\} \prod_{j=1}^{m} p_j^{n_j+a_j-1}$$

- with posterior estimates

$$\hat{\lambda} = E[\lambda | X^{(i)}] = \frac{N^i}{T_i}, \quad \hat{p}_j = E[p_j | X^{(i)}] = \frac{n_j^i+a_j}{N^i+a_0}$$

- Let $Z_j$ be arrival time of first type $j$ event

$$E[Z_j | X^{(i)}] = \frac{1}{\hat{\lambda} \cdot \hat{p}_j}$$
Estimating $C_{srch}()$

- Don’t observe $X_i$ but $Y_i$ where $Y_i = \min(X_1 ... X_i)$
- Must aggregate when new $Y_i$ is observed
- Changes posterior computation slightly

$$\hat{p}_j = E[p_j \mid X^{(i)}] = \frac{\tilde{n}_j + a_j}{N^i + a_0}$$
Estimating $C_{\text{comp}}()$

- Need fine analysis of computation time
- Model time as
  \[ t_{\text{comp}}(w) = \exp \{ \beta^T s \} \]
  where $s$ is a vector of features and $\beta$ are parameters
- Fit using standard regression analysis
- Ex. Design Matrix

\[
\begin{bmatrix}
1 & n_{v_1} & n_{f_1} & d_1 & s_1 & t_1 & w_1 \\
1 & n_{v_2} & n_{f_2} & d_2 & s_2 & t_2 & w_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & n_{v_M} & n_{f_M} & d_M & s_M & t_M & w_M
\end{bmatrix}
\]
Estimating $C_{\text{comp}}()$

Computation time of Bucket Elimination in different models

Actual and Estimated computation time of Bucket Elimination

- **Raw**
- **Predicted**
Stopping Criteria

- Current cost is: $C_i = T_i + t_{comp}(Y_i)$
- Predicted cost of width $j$ (denoted $W(j)$) is:
  \[
  E[C(W(j) | Y^{(i)})] = E[Z_j | Y^{(i)}] + t_{comp}(W(j))
  \]
- Let $j^i$ be index of largest $j$ such that $W(j) < Y_i$
- Stop when
  \[
  E[C(W(j)) | Y^{(i)}] \geq C_i
  \]
  for all $j < j^i$
- Let $i_{pred}$ denote the predicted termination index
Experiments

- Evaluation metrics:

\[ I_{err} = |i_{pred} - i_{best}| \quad \text{and} \quad T_{err} = |T_{i_{pred}} - T_{i_{best}}| \]

where \( i_{best} \) is the index where \( C_i \) is a minimum.

- Ex:

![Graph showing Total Cost - Search + Computation Time over iteration number]
Experiments

- lerr:

<table>
<thead>
<tr>
<th></th>
<th>1e-5</th>
<th>1e-4</th>
<th>1e-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree</td>
<td>6458 (4262.8)</td>
<td>4792.9 (4548.2)</td>
<td>2508.5 (3806.9)</td>
</tr>
<tr>
<td>largeFam3</td>
<td>3443.5 (3570.2)</td>
<td>4883.8 (3493.3)</td>
<td>6355.4 (3374.8)</td>
</tr>
<tr>
<td>largeFam4</td>
<td>2991.5 (3125.2)</td>
<td>3016.1 (3145.7)</td>
<td>4036.8 (3377.7)</td>
</tr>
<tr>
<td>largeFam5</td>
<td>3770.8 (3677.2)</td>
<td>2648.6 (3111.5)</td>
<td>3234.6 (3350.3)</td>
</tr>
</tbody>
</table>

- Terr

<table>
<thead>
<tr>
<th></th>
<th>1e-5</th>
<th>1e-4</th>
<th>1e-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree</td>
<td>93.69 (74.22)</td>
<td>73.55 (77.23)</td>
<td>36.64 (52.12)</td>
</tr>
<tr>
<td>largeFam3</td>
<td>868.0 (1343.0)</td>
<td>1023.2 (1292.4)</td>
<td>1148.9 (1238.1)</td>
</tr>
<tr>
<td>largeFam4</td>
<td>572.45 (892.9)</td>
<td>668.29 (1009.9)</td>
<td>817.1 (1023.9)</td>
</tr>
<tr>
<td>largeFam5</td>
<td>624.0 (988.9)</td>
<td>586.5 (1029.9)</td>
<td>730.2 (1135.4)</td>
</tr>
</tbody>
</table>
Alternate estimator of $t_{\text{comp}}$

- Compute ‘number of operations’ per ordering
- For BE, eliminating a variable in bucket $i$ involves $|S_i| \times |f_i|$ operations
  - where $f_i$ is the set of functions in bucket $i$
  - $S_i$ is the union of the scopes of $f_i$
- $\text{numOps}$ is the sum across all buckets
- Can easily compute $\text{numOps}$ given a heuristic
- Refer to this as $t_{\text{comp}}(\text{numOps})$
$t_{\text{comp}}(w)$ versus $t_{\text{comp}}(\text{numOps})$