Extensions of Dynamic Programming for Design and Analysis of Decision Trees

Mikhail Moshkov
King Abdullah University of Science and Technology
Saudi Arabia

Poland, December 2014
Research Group

Fawaz Alsolami  
Ph.D. Student, Computer Science  
Research Interest: Machine learning, Inhibitory Rules, Dynamic programming, Greedy algorithms  
fawaz.alsolami@kaust.edu.sa

Hassan AbouEisha  
Ph.D. Student, Computer Science  
Research Interest: Dynamic programming, Greedy algorithms, Finite element mesh solvers  
hassan.aboueisha@kaust.edu.sa

Mohammad Mohiuddin Azad  
Ph.D. Student, Computer Science  
Research Interest: Dynamic programming, Decision rules, Decision trees, Machine learning, Combinatorial machine learning, Greedy algorithms, Sequential optimization  
mohammad.azad@kaust.edu.sa

Shahid Hussain  
Ph.D. Student, Computer Science  
Research Interest: Discrete optimization, Dynamic programming, Decision trees, Machine learning, Greedy algorithms  
shahid.hussain@kaust.edu.sa

Talha Amin  
Ph.D. Student, Computer Science  
Research Interest: Dynamic programming, Decision rules, Greedy algorithms, Machine learning  
talha.amin@kaust.edu.sa
Research Group

Dr. Beata Zielosko, SRS
Abdulaziz Alkhalid, PhD student
Chandra Prasetyo Utomo, MS student with thesis
Enas Mohammad, MS student with thesis
Malek A. Mahayni, MS student with thesis

Maram Alnafie, Dir. Res.
Jewahir AbuBekr, Dir. Res.
Majed Alzahrani, Dir. Res.
Saad Alrawaf, Dir. Res.
Mohammed Al Farhan, Dir. Res.
Liam Mencel, Dir. Res.

Alumni

Monther Busbait
Dr. Igor Chikalov, Consultant
“Greatest Problem of Science Today”

• Tomaso Poggio and Steve Smale, The mathematics of learning: dealing with data, Notices of The AMS, Vol. 50, Nr. 5, 2003, 537-544

• The problem of understanding intelligence is said to be the greatest problem in science today and “the” problem for this century—as deciphering the genetic code was for the second half of the last one
Remark from KDnuggets


• While there is now a glut of industry and business oriented conferences on Big Data and Data Science, the technology which powers the current boom in Big Data comes from research ... (after that – a list of top research conferences in Data Mining, Data Science)
Dynamic Programming

- The idea of dynamic programming is the following. For a given problem, we define the notion of a sub-problem and an ordering of sub-problems from “smallest” to “largest”
- If (i) the number of sub-problems is polynomial, and (ii) the solution of a sub-problem can be easily (in polynomial time) computed from the solution of smaller sub-problems then we can design a polynomial algorithm for the initial problem
Dynamic Programming

- The aim of usual Dynamic Programming (DP) is to find an optimal object from a finite set of objects
Extensions of DP

We consider extensions of dynamic programming which allow us

- To describe the set of optimal objects
- To count the number of these objects
- To make sequential optimization relative to different criteria
- To find the set of Pareto optimal points for two criteria
- To describe relationships between two criteria
Extensions of DP

The areas of applications include
- Combinatorial optimization
- Finite element method
- Fault diagnosis
- Complexity of algorithms
- Machine learning
- Knowledge representation
Applications for Decision Trees

In the presentation, we consider applications of this new approach to the study of decision trees

• As algorithms for problem solving
• As a way for knowledge extraction and representation
• As predictors which, for a new object given by values of conditional attributes, define a value of the decision attribute
Decision Trees

Decision tree:

- Decision table:

<table>
<thead>
<tr>
<th>f_1</th>
<th>f_2</th>
<th>f_3</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Depth
Number of nodes
Total path length (average depth)
Number of terminal nodes

Cost functions
Directed Acyclic Graph $\Delta_0(T)$
Directed Acyclic Graph $\Delta_\alpha(T)$

<table>
<thead>
<tr>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$\Delta_\alpha(T)$

- $(f_2, 1)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 1     | 0     | 0     |
  - 0     | 1     | 0     |

- $(f_3, 1)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 1     | 0     | 0     |
  - 0     | 1     | 0     |

- $(f_3, 0)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 1     | 0     | 0     |
  - 0     | 1     | 0     |

- $(f_1, 0)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 0     | 1     | 0     |

- $(f_2, 0)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 1     | 0     | 0     |
  - 0     | 1     | 0     |

- $(f_1, 1)$
  - $f_1$ | $f_2$ | $f_3$ |
  - 1     | 1     | 1     |
  - 0     | 0     | 0     |
  - 1     | 0     | 0     |
  - 0     | 1     | 0     |
About Scalability

Table 1. Experimental results for Poker Hand data set

<table>
<thead>
<tr>
<th>sf</th>
<th>nodes</th>
<th>time</th>
<th>optimal</th>
<th>greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>depth</td>
<td>avg depth</td>
</tr>
<tr>
<td>0</td>
<td>1426236</td>
<td>177</td>
<td>5</td>
<td>4.08</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>1112633</td>
<td>124</td>
<td>5</td>
<td>3.99</td>
</tr>
<tr>
<td>$10^{-7}$</td>
<td>293952</td>
<td>27</td>
<td>4</td>
<td>3.73</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>79279</td>
<td>7</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>15395</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>4926</td>
<td>&lt; 1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>246</td>
<td>&lt; 1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>21</td>
<td>&lt; 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$10^{-1}$</td>
<td>1</td>
<td>&lt; 1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Training part of Poker Hand data set contains 25010 objects and 10 conditional attributes
Restricted Information Systems

• We described classes of decision tables for which the considered algorithms have polynomial time complexity depending on the number of conditional attributes
Extensions of DP for Decision Trees

- Sequential optimization
- Evaluation of the number of optimal trees
- Relationships between cost and accuracy
- Relationships between two cost functions
- Construction of the set of Pareto optimal points
Sorting of 8 Elements

• We proved that the minimum average depth of a decision tree for sorting 8 elements is equal to \( \frac{620160}{40320} \)

• This solved a long-standing problem (since 1968) considered by D. Knuth in his famous book *The Art of Computer Programming, Volume 3, Sorting and Searching*

• We proved also that each decision tree for sorting 8 elements with minimum average depth has minimum depth. The number of such trees is equal to \( 8.548 \times 10^{326365} \)
Corner points are used in computer vision for object tracking (FAST algorithm devised by Rosten and Drummond).

A pixel is assumed to be a *corner point* if at least 12 contiguous pixels on the circle are all either brighter or darker than the central point by a given threshold.
Corner Point Detection

Dynamic programming approach allows us to construct decision trees for corner point detection with average time complexity 7% less than for known ones, and analyze time-memory tradeoff for such trees.
Diagnosis of 0-1 Faults

Number $M(n)$ of monotone Boolean functions with $n$, $1 \leq n \leq 5$, variables.

<table>
<thead>
<tr>
<th>$n$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(n)$</td>
<td>3</td>
<td>6</td>
<td>20</td>
<td>168</td>
<td>7581</td>
</tr>
</tbody>
</table>
Diagnosis of 0-1 Faults

\[ h(S) \leq \begin{cases} 
(n + 1)L(S), & 1 \leq n \leq 4, \\
(n + 2)L(S), & n = 5.
\end{cases} \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H(n) )</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>( \varphi(n) )</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>
Table 1: The number of monotone boolean functions, $M(n)$, and the number of boolean functions, $B(n)$, with $n = 0, \ldots, 7$ variables.

<table>
<thead>
<tr>
<th>$n$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M(n)$</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>20</td>
<td>168</td>
<td>7581</td>
<td>7828354</td>
<td>2414682040998</td>
</tr>
<tr>
<td>$B(n)$</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>256</td>
<td>65536</td>
<td>$4.2 \times 10^9$</td>
<td>$1.8 \times 10^{19}$</td>
<td>$3.4 \times 10^{38}$</td>
</tr>
</tbody>
</table>
Table 2: The existence (example $f_i$) or nonexistence (—) of a monotone boolean function (MON) or a boolean function (ALL) with $n$ variables which does not have totally optimal decision trees relative to a subset of the set of parameters $\{D = \text{depth}, T = \text{total path length}, N = \text{number of nodes}\}$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>{D, N}</th>
<th>{D, T}</th>
<th>{T, N}</th>
<th>{D, T, N}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MON</td>
<td>ALL</td>
<td>MON</td>
<td>ALL</td>
</tr>
<tr>
<td>0</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>$f_3$</td>
</tr>
<tr>
<td>5</td>
<td>$f_1$</td>
<td>—</td>
<td>$f_3$</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>7</td>
<td>$f_5$</td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
<tr>
<td>$&gt;7$</td>
<td>$f_5$</td>
<td>$f_1$</td>
<td>$f_2$</td>
<td>$f_3$</td>
</tr>
</tbody>
</table>
Totally Optimal Decision Trees for Boolean Functions

\[
\begin{align*}
f_1 &= x_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 \lor \overline{x}_1 \overline{x}_2 x_3 \lor \overline{x}_1 x_3 x_5 \lor \overline{x}_1 x_4 \lor x_2 x_4 \lor x_3 x_4 x_5 \\
f_2 &= x_1 x_2 x_4 \lor x_1 x_4 x_5 \lor x_5 x_6 \lor x_3 x_4 \lor x_3 x_6 \\
f_3 &= \overline{x}_1 x_2 \overline{x}_4 \lor \overline{x}_1 x_3 x_4 \lor \overline{x}_2 \overline{x}_3 \\
f_4 &= x_1 \overline{x}_2 \overline{x}_3 x_5 \lor x_1 x_3 \overline{x}_4 \overline{x}_5 \lor x_1 x_4 x_5 \lor \overline{x}_1 x_2 x_3 x_5 \lor \overline{x}_1 \overline{x}_2 x_3 \overline{x}_5 \\
&\quad \lor \overline{x}_1 \overline{x}_2 \overline{x}_3 x_4 \lor \overline{x}_1 x_4 \overline{x}_5 \lor x_2 \overline{x}_3 x_4 \overline{x}_5 \\
f_5 &= x_1 x_2 x_5 x_7 \lor x_1 x_2 x_6 x_7 \lor x_1 x_3 x_6 x_7 \lor x_1 x_4 x_6 x_7 \lor x_2 x_3 x_6 x_7 \\
&\quad \lor x_2 x_5 x_6 x_7 \lor x_1 x_4 x_5 \lor x_2 x_4 x_5 \lor x_3 x_4 x_5
\end{align*}
\]
Heuristics for Decision Tree Construction

Minimization of decision tree average depth for decision tables with many-valued decisions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ws_entML</td>
<td>3.26%</td>
</tr>
<tr>
<td>ws_entSort</td>
<td>3.49%</td>
</tr>
<tr>
<td>Div_ws_entSort</td>
<td>4.53%</td>
</tr>
</tbody>
</table>
Minimization of Number of Nodes

Decision table *Mushroom* contains 22 conditional attributes and 8124 rows.

The minimum number of nodes in a decision tree for *Mushroom* is equal to 21.
When the number of misclassifications is increasing, the number of nodes in decision trees can decrease.

One can be interested in less accurate but more understandable decision trees.

Tic Tac Toe, 9 attributes, 959 rows
Decision Trees and Rules

- Decision rules are widely used in machine learning and for knowledge representation
- One of the ways to obtain decision rules is to construct a decision tree and derive rules from this tree

Decision tree

\[ f_1 = 0 \land f_2 = 0 \rightarrow d = 3 \]
\[ f_1 = 0 \land f_2 = 1 \rightarrow d = 2 \]
\[ f_1 = 1 \rightarrow d = 1 \]

Set of decision rules
Relationships Depth vs. Number of Terminal Nodes

Lymphography, 18 attributes, 148 rows

Nursery, 8 attributes, 12960 rows
Relationships between the number of nodes and the number of misclassifications can be used in a special procedure of pruning.

Breast cancer, 9 attributes, 266 rows
Three Books Published by Springer

“Bridge” among three approaches in Data Analysis which previously were not connected

Textbook for the course CS361 in KAUST

Research monograph
New Book and New Course

Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining
KAUST

• KAUST is an international graduate-level research university located on the shores of the Red Sea in Saudi Arabia

• The University’s new facilities, excellent faculty, state-of-art library and Shaheen Supercomputer offer an ideal environment and resources for graduate level study and research
KAUST

Students receive a KAUST fellowship that includes:

• full tuition
• competitive monthly living allowance
• private medical and dental coverage
• housing
• relocation support
KAUST