Neither Basic Nor Applied
Lessons From a Computer Research Journey

Peter Lee
Microsoft Research Redmond
F(n) = if n == 0 then return 1 else return n * F(n-1)

F = “the factorial function”
   is a solution...

...but it is also a solution for

F(n) = if n == 0 then return 1 else return F(n+1) / (n+1)
“If you think that beauty is so important, maybe you should be in the School of Art instead of the Computer Science Department.”

— a former CMU CS Department Head, speaking to Peter Lee, Assistant Professor
“Things like even software verification, this has been the Holy Grail of computer science for decades but now in some very key areas, for example, driver verification we’re building tools that can do actual proof about the software and how it works in order to guarantee the reliability.”

_ Bill Gates, April 18, 2002_
F(n) =

if n == 0 then return 1
else return n * F(n-1)
open-ended

reactive

short-term

long-term

Disruptive

Blue sky

Mission-focused

Sustaining

“Mudge’s quadrants”
open-ended

reactive

short-term

Disruptive

Mission-focused

long-term

Blue sky

Sustaining
open-ended

reactive

short-term

Mission-focused

long-term
Rehospitalizations among Patients in the Medicare Fee-for-Service Program

Stephen F. Jencks, M.D., M.P.H., Mark V. Williams, M.D., and Eric A. Coleman, M.D., M.P.H.

ABSTRACT

Background Reducing hospital readmissions is a way to improve quality of care, reduce the frequency and preventable complications, and lower unnecessary changes to hospitals.

Methods We analyzed discharge records for Medicare patients to identify the causes of hospitalizations and to calculate the annualized cost to Medicare in 2004.

Results Almost one third of Medicare beneficiaries were discharged from a hospital within 30 days, and 34.0% were rehospitalized within 90 days. 37.6% of patients who had been discharged within 30 days were rehospitalized within 90 days. The estimated cost to Medicare in 2004 was $17.4 billion.

- ~20% within 30 days
- ~35% in 90 days
- Estimated cost to Medicare in 2004: $17.4 billion
<table>
<thead>
<tr>
<th>Weight</th>
<th>Feature description</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.68398</td>
<td>Dx0-&gt;2 = Excessive vomiting in pregnancy</td>
<td>0.31%</td>
</tr>
<tr>
<td>0.61306</td>
<td>Dx3-&gt;2 = Personal history of malignant neoplasm</td>
<td>0.28%</td>
</tr>
<tr>
<td>0.58281</td>
<td>Dx0-&gt;2 = Heart failure</td>
<td>0.30%</td>
</tr>
<tr>
<td>0.56708</td>
<td>Dx0-&gt;1 = Nephritis, nephrotic syndrome, and nephrosis</td>
<td>0.09%</td>
</tr>
<tr>
<td>0.56649</td>
<td>Dx3-&gt;2 = Heart failure</td>
<td>0.28%</td>
</tr>
<tr>
<td>0.54663</td>
<td>Complaint sentence contains &quot;suicidal&quot;</td>
<td>0.17%</td>
</tr>
<tr>
<td>0.48415</td>
<td>Dx1-&gt;2 = Disorders of function of stomach</td>
<td>0.07%</td>
</tr>
<tr>
<td>0.47257</td>
<td>Dx5-&gt;0 = Diseases Of The Genitourinary System</td>
<td>0.15%</td>
</tr>
<tr>
<td>0.46136</td>
<td>Dx0-&gt;2 = Chronic airway obstruction, not elsewhere classified</td>
<td>0.10%</td>
</tr>
<tr>
<td>0.44555</td>
<td>Dx4-&gt;2 = Depressive disorder, not elsewhere classified</td>
<td>0.10%</td>
</tr>
<tr>
<td>0.44257</td>
<td>Stayed 14 hours in the ER</td>
<td>0.10%</td>
</tr>
<tr>
<td>0.43890</td>
<td>Dx0-&gt;1 = Other psychoses</td>
<td>0.32%</td>
</tr>
<tr>
<td>0.43513</td>
<td>Dx0-&gt;0 = Diseases Of The Blood And Blood-Forming Organs</td>
<td>0.46%</td>
</tr>
<tr>
<td>0.42582</td>
<td>Complaint sentence contains &quot;dialysis&quot;</td>
<td>0.19%</td>
</tr>
<tr>
<td>0.41888</td>
<td>Dx0-&gt;2 = Depressive disorder, not elsewhere classified</td>
<td>0.27%</td>
</tr>
<tr>
<td>0.41302</td>
<td>Dx1-&gt;1 = Nephritis, nephrotic syndrome, and nephrosis</td>
<td>0.99%</td>
</tr>
<tr>
<td>0.38506</td>
<td>Complaint sentence contains &quot;fluid&quot;</td>
<td>0.10%</td>
</tr>
<tr>
<td>0.37474</td>
<td>69 &lt; Age</td>
<td>9.22%</td>
</tr>
</tbody>
</table>
Learning to Rank

Chris J. C. Burges
Microsoft Research
Learning to rank - Wikipedia, the free encyclopedia
Applications · Feature vectors · Evaluation measures · Approaches
Learning to rank or machine-learned ranking (MLR) is a type of supervised or semi-supervised machine learning problem in which the goal is to automatically construct a ranking...
en.wikipedia.org/wiki/Learning_to_rank

Yahoo! Learning to Rank Challenge
Learning to Rank Challenge is closed! Close competition, innovative ideas, and fierce determination were some of the highlights of the first ever Yahoo!
learningtorankchallenge.yahoo.com

Learning to Rank using Gradient Descent
Learning to Rank using Gradient Descent Keywords: ranking, gradient descent, neural networks, probabilistic cost functions, internetsearch Chris Burges cburges@microsoft.com ... research.microsoft.com/en-us/um/people/cburges/papers/ICML_ranking.pdf · PDF file
How LambdaRank is Trained

Query: “107.7 the end”

<table>
<thead>
<tr>
<th>URL</th>
<th>Rating</th>
<th>Model Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.1077theend.com">http://www.1077theend.com</a></td>
<td>Perfect</td>
<td>0.13</td>
</tr>
<tr>
<td><a href="http://en.wikipedia.org/wiki/107.7_The_End">http://en.wikipedia.org/wiki/107.7_The_End</a></td>
<td>Good</td>
<td>0.21</td>
</tr>
<tr>
<td><a href="http://www.myspace.com/1077theend">http://www.myspace.com/1077theend</a></td>
<td>Good</td>
<td>0.19</td>
</tr>
<tr>
<td><a href="http://en.wikipedia.org/wiki/The_End">http://en.wikipedia.org/wiki/The_End</a></td>
<td>Fair</td>
<td>0.01</td>
</tr>
<tr>
<td><a href="http://www.thewolf.co.uk/">http://www.thewolf.co.uk/</a></td>
<td>Bad</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Cost = (score1 - score2)

<table>
<thead>
<tr>
<th>y_1</th>
<th>y_2</th>
<th>y_3</th>
<th>y_4</th>
<th>y_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.525</td>
<td>-0.040</td>
<td>0.038</td>
<td>0.551</td>
<td>-0.023</td>
</tr>
</tbody>
</table>
# Building an ensemble of regression trees

<table>
<thead>
<tr>
<th>Rating</th>
<th>Url</th>
<th>Score</th>
<th>Force</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td><a href="http://en.wikipedia.org/wiki/107.7_The_End">http://en.wikipedia.org/wiki/107.7_The_End</a></td>
<td>0.20</td>
<td>-0.09</td>
</tr>
<tr>
<td>Perfect</td>
<td><a href="http://www.1077theend.com">http://www.1077theend.com</a></td>
<td>0.19</td>
<td>0.37</td>
</tr>
<tr>
<td>Good</td>
<td><a href="http://www.myspace.com/1077theend">http://www.myspace.com/1077theend</a></td>
<td>0.19</td>
<td>0.04</td>
</tr>
<tr>
<td>Bad</td>
<td><a href="http://www.thewolf.co.uk/">http://www.thewolf.co.uk/</a></td>
<td>0.17</td>
<td>-0.35</td>
</tr>
<tr>
<td>Fair</td>
<td><a href="http://en.wikipedia.org/wiki/The_End">http://en.wikipedia.org/wiki/The_End</a></td>
<td>0.01</td>
<td>0.10</td>
</tr>
</tbody>
</table>

\[ \text{score} = s_0 + \]

\[ \text{score} \] = \text{sum of the scores of individual trees}
### How is a split computed?

<table>
<thead>
<tr>
<th>Rating</th>
<th>Url</th>
<th>BM25F</th>
<th>Count of query in body</th>
<th>Anchor text matches query</th>
<th>Function(click frequency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad</td>
<td><a href="http://www.thewolf.co.uk/">http://www.thewolf.co.uk/</a></td>
<td>55370</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Good</td>
<td><a href="http://www.myspace.com/1077theend">http://www.myspace.com/1077theend</a></td>
<td>80981</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Perfect</td>
<td><a href="http://www.1077theend.com">http://www.1077theend.com</a></td>
<td>81023</td>
<td>60</td>
<td>25286</td>
<td>251</td>
</tr>
<tr>
<td>Fair</td>
<td><a href="http://en.wikipedia.org/wiki/The_End">http://en.wikipedia.org/wiki/The_End</a></td>
<td>80984</td>
<td>156</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
BM25F \leq 81000
\]

\[
Loss = \sum_i (y_i - \bar{y})^2
\]

\[
Loss = \sum_{i \in L} (y_i - \bar{y}_L)^2 + \sum_{i \in R} (y_i - \bar{y}_R)^2
\]
FastRank: Algorithms ↔ Engineering

Minimizing the loss is equivalent to maximizing \( \frac{1}{|L|} (\sum_{i \in L} y_i)^2 + \frac{1}{|R|} (\sum_{i \in R} y_i)^2 \).

We only need one linear scan of the data (in a given node) to compute this.
open-ended

reactive

short-term

Mission-focused

Sustaining

long-term
Microsoft Translator

Daily Translation Requests

30x growth in 2 years
Whitebox Fuzzing

David Molnar
Microsoft Research
Problem: Security Bugs in File Parsers
Ongoing challenge for Microsoft ecosystem

Hundreds of file formats supported in Windows, Office, etc.


Every security patch costs Microsoft alone one million dollars.

Traditional random fuzz testing can’t catch this bug:

```c
int obscure(int x, int y) {
    if (x==hash(y)) error();
    return 0;
}
```
Approach: “Whitebox” Fuzz Testing
State of the art solving + industrial strength binary analysis

```c
void top(char input[4])
{
    int cnt = 0;
    Path constraint:
    if (input[0] == 'b') cnt++;
    if (input[1] == 'a') cnt++;
    if (input[2] == 'd') cnt++;
    if (input[3] == '!') cnt++;
    if (cnt >= 3) crash();
}
```

SAGE originated in Center for Software Excellence (2006-2008)
SAGE and SAGAN
Impact: Change Security Testing

Largest deployment of solvers in the world

Fuzzing bugs found in Win7 “WEX” over 100s of file parsers:

- Run in every Win8 milestone
- Ongoing engagement with Office and Microsoft Security Engineering Center
- 200+ machine years
- One billion+ constraints

Big thank-you to our partners across Microsoft!
Research: Feedback From Scale
Any test anywhere helps all future tests everywhere!

How much **sharing** between symbolic execution of **different** programs run on Windows?

Most common branch appears **17761** times out of **290430** symbolic executions. Motivates **symbolic summaries built up over time**.

How does the Z3 solver perform on **constraints** arising from **real code**?

90.18% of Z3 queries solved in **0.1 seconds or less**. Solving time still dominates! Tells us where to focus Z3.

Leverage data collection to create **virtuous cycle** that keeps us ahead of the competition!

Answer questions and pursue directions **impractical without scale**.
Thanks to all SAGE contributors!

- MSR: Ella Bounimova, Patrice Godefroid, David Molnar
  (+ our managers for their support! 😊)
- CSE: Michael Levin, Chris Marsh, Lei Fang, Stuart de Jong,...
- Interns: Dennis Jeffries (06), David Molnar (07), Adam Kiezun (07), Bassem Elkarablieh (08), Marius Nita (08), Cindy Rubio-Gonzalez (08,09), Johannes Kinder (09), Daniel Luchaup (10), Mehdi Bouaziz (11),...
- Z3 (MSR): Nikolaj Bjorner, Leonardo de Moura,...
- Windows: Nick Bartmon, Eric Douglas, Dustin Duran, Elmar Langholz, Isaac Sheldon, Dave Weston,...
  - Win8 TruScan support: Evan Tice, David Grant,...
- Office: Tom Gallagher, Eric Jarvi, Octavian Timofte,...
- MSEC: Dan Margolis, Matt Miller, Lars Opstad, Jason Shirk,...
- SAGE users all across Microsoft!
- Download SAGE (Microsoft only): http://sharepoint/sites/SAGE
open-ended
reactive
short-term
long-term

Mission-focused
Sustaining
Blue Sky
Topological Quantum Systems

Fractional Quantum Hall States

Topological Superconductors

Spin Systems

Topological devices are immune to local errors,
Thus relieving much of the burden of error correction.
Programming on the Phone

Nikolai Tillmann
Microsoft Corporation
Program smartphones directly!

- In 2010, smartphones outsold PCs
- Today’s smartphones are more powerful than PCs from 2000
- How do we use them?
DEMO

Programming on the Phone
Programming directly on the phone...

- Is popular
  - Top 200 of Windows Phone Marketplace; top 10 of Productivity apps
  - Rated 4.7/5 by users

- Is great for teaching
  - Engaging programming experience
  - Using personal device and data (songs, pictures, etc.)

- Embraces new reality
  - Language and editor optimized for touchscreen
  - First-class access to sensors and services
  - Sharing and learning with friends in the cloud
open-ended

reactive

short-term
long-term

Disruptive
Mission-focused

Blue Sky
Sustaining
Audio for Kinect

Ivan Tashev
Microsoft Research
open-ended
reactive

Disruptive
Mission-focused

Blue sky
Sustaining

short-term
long-term
Computing research...

...has incredible diversity,

and rarely is exclusively “basic” or “applied”.
Thank you!

petelee@microsoft.com