Experimental Asymptotic Analysis of Algorithms

NISS
Catherine C. McGeoch
March 2008
Algorithm = Abstraction

Algorithm

Quicksort A:
- Select element x from array A: constant cost
- Partition A around x: linear cost
- Recur to left of x
- Recur to right of x

Program/Code

void Qsort(A, l, h) {
    if (l >= h) return;
    int p = Partition(A);
    Qsort(A, l, p-1);
    Qsort(A, p+1, h);
}

Running Process

Measure this

Draw conclusions about this
Analyzing an Algorithm

For a given input space:

How much time will the algorithm take, as a function of $n$?

How close to optimal is the output it produces, as a function of $n$?
Asymptotic Analysis

Definition: A function \( f(n) \) is in the set \( O(g(n)) \) if there exist constants \( c > 0 \) and \( n_0 > 0 \) such that

\[
0 \leq f(n) \leq c \cdot g(n) \quad \forall n > n_0.
\]

What is the order of the leading term of the function? What is an upper (lower) bound on the order of the leading term?

\[
\begin{align*}
f(n) = 3n^2 - 6n + 12 & \text{ is } O(n^2) \\
f(n) = 0.5n + \log_2 n & \text{ is } O(n) \\
f(n) = 20 \log_2 n + 4 & \text{ is } O(\log n) \\
f(n) = 3500 \cdot 7/n & \text{ is } O(1)
\end{align*}
\]
Asymptotic Curve Bounding

Curve fitting = find a curve that is close to the observed data within the range of observations.

Curve bounding = find a curve that you are confident is an upper (lower) bound on the data, even beyond the range of observations.
Why Asymptotic Algorithm Analysis?

• Dominant cost model explains / predicts performance best when $n$ is large.

• We care more about cost when $n$ is large.

• Death, taxes, problem sizes: $n$ will be larger in the future.

• Asymptotic properties are universal, fundamental, and independent of transient technology (platforms, programming languages, coding skills).
Average-Case Analysis

• **Input:** Draw instances of size $n$ at random from parameterized space $S(m, k, \ldots)$.

• **Experiment:** Measure algorithm performance in several independent trials for varying $n, m, k\ldots$

• **Goal:** Find an asymptotic function $C_{m,k}(n)$ that bounds the mean cost (Time or Solution Quality).
## Experiments on Algorithms

<table>
<thead>
<tr>
<th><strong>Good news</strong></th>
<th><strong>Bad news</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearly total control over the experiment.</td>
<td>Unusual data: skewed, bounded, nonmonotonic, stepped.</td>
</tr>
<tr>
<td>Algorithms are easy to probe.</td>
<td>Unusual questions: Asymptotic analysis.</td>
</tr>
<tr>
<td>Simple mechanisms, models (compared to living things).</td>
<td>Unusual questions: Curve bounding vs curve fitting.</td>
</tr>
<tr>
<td>Lots of data points, usually.</td>
<td>Unusually precise questions: is it $O(n)$ or $O(n \log n)$?</td>
</tr>
<tr>
<td>Model validation not much of a problem.</td>
<td></td>
</tr>
</tbody>
</table>
Outline

• Three Case Studies in Algorithm Research
  – FF Rule for Bin Packing
  – All Pairs Shortest Paths with Essential Subgraph
  – Sampling Graph Colorings

• Some Data Analysis "Techniques" I've Tried
  – Power Law
  – Guessing
  – Data Transformation
  – Others

• My Questions, Your Questions
Three Case Studies, Many Questions

- FF Rule for Bin Packing
- All Pairs Shortest Paths with the Essential Subgraph
- Sampling Graph Colorings with Jerrum's algorithm

- How do I analyze the data to find asymptotic bounds?
- How do I assess the quality of my analysis? How confident am I in the results?
- Where do I place sample points? How many random trials?
- How do I design my second experiment?
- Which performance measures are easier to analyze? How can I tell in advance?
First Fit (FF) Bin Packing

Input: List of \( n \) item sizes drawn uniformly iid from \((0,u), \ 0 < u \leq 1\).

First Fit Algorithm: Pack items into unit-sized bins

Solution Quality: How much empty space in the packing?
For given $u$, mean empty space $e_u(n)$ is either asymptotically linear or strictly sublinear in $n$. Sublinearity implies optimality.

For which values of $u$ is $e_u(n)$ optimal?
Empty Space at N = 8 million

Some values of \( u \) produce bad FF packings. How bad? Which values of \( u \)?
**Empty Space growth in n**

Power law: Linear regression on log-log scale. Analyze slope: If $e = an^b$ then $\log e = b \log n + \log a$

$u=1$ appears to be sublinear, slope near 0.68.

Others have slopes in $(0.974 \ldots 0.998)$. Are they asymptotically 1 (linear)?
**Sublinear when u=1. What function?**

Guess the leading term is of the form $cn^{2/3}$, plot $e/n^{2/3}$, assess convergence to a constant.

Is $e$ asymptotically $O(n^{2/3})$ or $O(n^{2/3} \log n)$?

Is this function bounded above by a constant?
All Pairs Shortest Paths (APSP)

Input: Complete graphs $G$, on $n$ vertices, with weights on edges iid uniform from $(0,1)$.

Algorithm computes APSP using subgraph $H$.

Time depends on $H$: How many edges in $H$?

Input $G$, $n=5$

APSP: all vertex-pair distances
S edges in $H$: $O(n)$ or $O(n \log n)$?

Known: $n-2 < s$ and $E[s] < 13.5 \ n \ log_e n$

Plot: $S/n \ log n$.

Does this converge to 0 or to $c>0$?

What is the asymptotic lower bound on $c$?
$S$: $O(n)$ or $O(n \log n)$?

Plot: $S/n$. Does this converge to $c > 0$? Or does it grow unbounded by a constant?
What is the rank $R$ of the largest edge in $H$ among the $n(n-1)/2$ edges in $G$?

*Known*: $S \leq R$ and $n \log_e n < E[R]$

Plot of $n$ vs $R/S$. Does this converge to a constant $c$? What is an upper bound on $c$?
Size vs Rank

Plot of $S$ vs $R$. How can I bound asymptotically the mean and the expected max value of $R$?
Jerrum's Graph Coloring Sampling Algorithm

Input: Grid graph \( G \) of \( n \) vertices, degree \( d \) in \((4,6,8)\), and a color count \( k \).

Jerrum's Algorithm: random walk in space of colorings

Output \( C \): A valid coloring of \( G \), drawn uniformly from the space of valid \( k \)-colorings.

Time: How quickly does the distribution of the random walk converge to (within \( \varepsilon \) of) uniform?
Jerrum’s Algorithm

Theorem: For any graph G, n nodes, maximum degree d, color set k:

• If \( k \geq 2d \) the algorithm converges to Uniform in polynomial time.
• If \( k = d+1 \) the algorithm takes exponential time to converge.
• If \( k \leq d \) the algorithm does not converge.

What about \( k \) in the range \((d+2, 2d-1)\)?

Conjecture: exponential throughout.

*Time to couple* is an upper bound on convergence rate. Proofs are especially difficult for *grid graphs*.....
Jerrum's Algorithm: Coupling Time

Time to Couple, $T$, is an upper bound on time to converge. Three trials, $n=64$, $d=8$, $k=12$. 
Coupling Time

Grid graph
\(d=8,\)
\(k=(9..17),\)
\(n=(36, 64, \ldots 144),\) 50 trials; note cutoff at 500000.
Coupling Time

Grid graph $d=8$, $k=(9..17)$, $n=(36, 64, ... 144)$. Means of 50 trials; note cutoff.

For which $k$ does $T$ show exponential growth in $n$?
Log coupling time means of 50 trials.

d=8, k=9: known exponential.

d=8, k=17: known polynomial.

How do I classify the others? Where is the critical point?
Coupling Times for Grid Graphs

Log coupling times, means of 50 trials.

\(d=6, k=7\): known exponential.

\(d=6, k \geq 12\): known polynomial.

How do I classify the others? Where is the critical point?
Questions

• Bin Packing: Convergence of empty space (a difference) is easier to evaluate than convergence of bin counts (a ratio). Why?

• Is R (rank of largest edge) easier to analyze than S (number of edges)? How to find an asymptotic upper bound on the expected maximum?

• Jerrum's algorithm: How to distinguish polynomial from exponential functions?

• Sampling: Is an experiment with 1000 N values evenly spaced between 1 and N_max easier to evaluate than one with 10 points each at N, N/2, N/4, N/8 ...? Why?
Where to place sample points?

Empty space as $f(u)$

$\ n=128k$  

ESL as $f(u)$

$\ n=8m$
Three Case Studies, Many Questions

• FF Rule for Bin Packing
• All Pairs Shortest Paths with the Essential Subgraph
• Sampling Graph Colorings with Jerrum's algorithm

• How do I analyze the data to find asymptotic bounds?
• How do I assess the quality of my analysis? How confident am I in the results?
• Where do I place sample points? How many random trials?
• How do I design my second experiment?
• Which performance measures are easier to analyze? How can I tell in advance?
Asymptotic Curve Bounding

Generated data: Is y growing linearly, quadratically, or somewhere in between? Find an upper or lower bound.
Some Asymptotic Curve Bounding Techniques

- Power Law
- Guess - Ratio
- Guess - Difference
- Box - Cox transformation
- Newton's method of differences
- Generalized regression
- Tukey's ladder of transformations
Power Law

1. Plot log-log data.
2. Fit a line.
3. Check slope.
4. Check residuals.
Residuals from Power Law Fit

Conclusion: $y$ grows faster than $x^{1.02}$
Guess - Ratio

1. Guess a function $g(x)$.
2. Plot $y/g(x)$.
3. If increasing: $y$ grows faster than $g(x)$.
4. If decreasing to 0: $y$ grows slower than $x$.
5. If converging to constant $> 0$: $y$ grows as $x$.
Conclusion (from iterated guesses): $y$ grows faster than $x^{1.1}$.

(Slower than $x^2$ ?)
1. Guess the first term $g(n) = an^b$.

2. Plot $g(n) - Y$: If down-up, $g(n)$ is an upper bound.

3. Iterate guess to find a tighter upper bound $g(n)$.

Conclusion: $y$ grows more slowly than $x^2$. 
Box-Cox Rule

1. Transform $y$ using $y^t$ (with scaling function).
2. Compare transformed data to a straight line.
3. Use scaled RSS to assess fit to line.
4. Repeat, find $t$ with min scaled RSS.
5. Invert $t$ to find $y$ as $f(x)$
Residuals from Box Cox Fit

Conclusion: $y$ grows more slowly than $x^{1.4}$
Newton's Method of Differences

1. Evaluate polynomial \( f(x) \) at evenly spaced \( x_1, x_2, x_3, \ldots, x_n \)
2. Find differences in adjacent evaluations.
3. Repeat until differences are constant.
4. Number of repetitions = degree of polynomial.

\[
\begin{array}{cccccc}
43 & 123 & 243 & 403 & 603 \\
80 & 120 & 160 & 200 \\
40 & 40 & 40 \\
\end{array}
\]

quadratic!

Problems:
- Only works on integer degree polynomials.
- Requires evenly spaced \( x \) values
- Can't cope with random data. No answer for this problem.
Generalized Regression

1. Guess a multi-term function \( g(n) \).
2. Iterate: add a term, delete a term ...
3. Use residuals, RSS to evaluate fit.
4. Find best fit, look at the leading term.

Problems:

- Best fit to the curve does not imply best choice of leading term.
- Different iteration methods (insert/delete paths) give different "best" fits. No sense of convergence to an optimal fit -- need an alternative to RSS.
- Residuals analysis can give contradictory results: growing faster than \( x^a \) and also growing slower than \( x^a \).
- *It doesn't work.*
Digression

• Can computer science help build a better generalized regression method? Current practice seems to be hill climbing with bad neighborhood rule and sketchy objective function.
Tukey's Transformation Ladder

1. Transform $y$ according to a scale (ladder) of choices:
   - $y^2$
   - $y^{1/2}$
   - $\log y$
   - $1/y$
   - $1/y^2$

2. Look for a straight line. If $\sqrt{y}$ is straightest, conclude $y = x^2$.

3. Or transform $x$, or transform both.

Problems:

- Transforming $x$ can give answers that contradict transforming $y$: $y$ is faster than $x^a$, and $y$ is slower than $x^a$.
- Low order terms have different importance in the transformed space.
- *It doesn't work.*
Asymptotic Curve Bounding

The answer: \( y = 3 \cdot x^{1.8} + 1000 \cdot x + 1000 + noise \)

- \( \checkmark \) **Power law**: \( y \) faster than \( x^{1.02} \)
- \( \checkmark \) **Guess - Ratio**: \( y \) faster than \( x^{1.1} \)
- \( \checkmark \) **Guess - Difference**: \( y \) slower than \( x^2 \)
- \( \times \) **Box - Cox**: \( y \) slower than \( x^{1.4} \)
- **(no answer)** *Newton's method of differences*:
Tests on Generated and Real Data

- **PW**: Power Law
- **PW3**: Power Law high 3 data points
- **PWD**: Power Law with differencing
- **GR**: Guess - Ratio
- **GD**: Guess - Difference with up/down heuristic
- **BC**: Box Cox
- **DF**: Newton's Differencing with "almost flat" heuristic

Functions $y = ax^b + cx^d$ varying $a, b, c, d$. Find a bound on $b$.

Functions $y = ax^b + cx^d + r$ with noise variate $r$.

Functions from algorithm research (some ranges known).

How much does increasing $x$ help?

How much does random noise hurt?

Can humans do better?
## Nonrandom Functions

<table>
<thead>
<tr>
<th>Expression</th>
<th>bc</th>
<th>pwd</th>
<th>gd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3x^{.2} + 1$</td>
<td>0.127</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$3x^{.2} + 10^2$</td>
<td>0.2</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>$3x^{.2} + 10^4$</td>
<td>0.2</td>
<td>24</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>pwd</th>
<th>gd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3x^{.8} + 10^4$</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td>$3x^{.8} + x^{.2}$</td>
<td>0.793</td>
<td>1</td>
</tr>
<tr>
<td>$3x^{.8} - x^{.2}$</td>
<td>x</td>
<td>0.807</td>
</tr>
<tr>
<td>$3x^{.8} + x^{.6}$</td>
<td>0.778</td>
<td>1</td>
</tr>
<tr>
<td>$3x^{.8} - x^{.6}$</td>
<td>x</td>
<td>0.829</td>
</tr>
<tr>
<td>$3x^{.8} + 10^4 x^{.6}$</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>$3x^{.8} - 10^4 x^{.6} + 10^6$</td>
<td>x</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>pwd</th>
<th>gd</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3x^{1.2} + 10^4$</td>
<td>1.2</td>
<td>1.22</td>
</tr>
<tr>
<td>$3x^{1.2} + x^{.2}$</td>
<td>1.19</td>
<td>1.2</td>
</tr>
<tr>
<td>$3x^{1.2} + 10^4 x^{.2}$</td>
<td>0.263</td>
<td>x</td>
</tr>
<tr>
<td>$3x^{1.2} + x$</td>
<td>1.175</td>
<td>1.21</td>
</tr>
<tr>
<td>$3x^{1.2} - x$</td>
<td>x</td>
<td>1.233</td>
</tr>
<tr>
<td>$3x^{1.2} + 10^4 x$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Tightest bounds found.

\[ x = 8, 16, 32, 64, 128 \]
Nonrandom Functions

3x^{0.2} + 1
3x^{0.2} + 10^2
3x^{0.2} + 10^4 \quad bc \quad NA

3x^{0.8} + 10^4 \quad bc \quad NA
3x^{0.8} + x^{0.2}
3x^{0.8} - x^{0.2} \quad gr \quad .825 \; lb
3x^{0.8} + x^{0.6}
3x^{0.8} - x^{0.6} \quad gr \quad .838 \; lb, \; bc \quad .819 \; lb
3x^{0.8} + 10^4 \times 0.6
3x^{0.8} - 10^4 \times 0.6 + 10^6 \quad pw, \; pw3, \; df
\quad negative/zero \; ub; \; pwd, \; bc \quad NA

3x^{1.2} + 10^4 \quad bc \quad NA
3x^{1.2} + x^{0.2}
3x^{1.2} + 10^4 \times 0.2 \quad gd \quad NA, \; df \quad 1 \; ub
3x^{1.2} + x
3x^{1.2} - x \quad gr \quad 1.238 \; lb, \; bc \quad 1.228 \; lb
3x^{1.2} + 10^4 \times 1ub

Wrong answers (bad bounds shown) and no answers (NA).

BC fails on nearly constant data (transformation \( y^{1/b} \) is undefined if \( b=0 \)).

GR fails on negative second order terms

DF ``almost flat'' rule can be fooled

All can fail on decreasing data, large second terms
**Data From Algorithms Research**

<table>
<thead>
<tr>
<th>What is known:</th>
<th>wrong/NA</th>
<th>lower ... upper bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = (x+1)(2H_{x+2} -2)$</td>
<td>gr, pwd</td>
<td>$x \ldots 1.18$ pw3</td>
</tr>
<tr>
<td>$y = (x^2 - x) / 4$</td>
<td>pwd</td>
<td>gr 2 \ldots 3.001 pw3</td>
</tr>
<tr>
<td>$E[y] = x/2 + O(1/x^2)$</td>
<td></td>
<td>gr,pw .99 \ldots x</td>
</tr>
<tr>
<td>$E[y] = \Theta(x^{1/2})$</td>
<td>gr</td>
<td>$x \ldots .5716$ pw3</td>
</tr>
<tr>
<td>$E[y] = O(x^{2/3} (\log x)^{1/2})$</td>
<td>gr</td>
<td>$x \ldots .695$ pw3</td>
</tr>
<tr>
<td>$E[y] = \Omega(x^{2/3})$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E[y] \leq 0.68 x$</td>
<td>pwd</td>
<td>pw .954 \ldots 1 gd,df</td>
</tr>
<tr>
<td>$x-1 \leq y \leq 13.5 x \log_e x$</td>
<td>gr, pw3, pwd</td>
<td>$x \ldots 1.142$ pw</td>
</tr>
<tr>
<td>$x \log_e x &lt; y &lt; 1.2 x^2$</td>
<td>pwd</td>
<td>gr 1.3 \ldots 1.31 pw</td>
</tr>
</tbody>
</table>

Note: Many rules failed to decide if the bound was upper or lower: returned ``close``. A close fit is bad in this context.
Some Conclusions

- **Power Law**
  Every rule sometimes fails.
- **Power Law Top 3**
  Generalized regression & Tukey's Ladder are not internally consistent. Contradictory answers are artifact of application.
- **Power Law with differencing**
  Doubling the largest problem size is less effective than expected: no rule ``became correct,'' and only a few have slightly tighter bounds.
- **Guess - Ratio**
- **Guess - Difference**
- **Box Cox**
- **Newton's Differencing**
- **Generalized regression**
- **Tukey's Ladder**
  Randomness in data makes curves in residuals harder to find; more ``close" answers, fewer ``upper/lower bound" answers.
  Humans do about as well as automated rules, but much more slowly.
More Questions

- **Power Law**
- **Power Law Top 3**
- **Power Law with differencing**
- **Guess - Ratio**
- **Guess - Difference**
- **Box Cox**
- **Newton's Differencing**
- **Generalized regression**
- **Tukey's Ladder**

How to cope with logarithms in terms?

When/why should I trust the answer returned by the rule?

Can generalized regression & Tukey's Ladder be fixed?

I can't always choose whether the rule returns an upper bound or lower bound. Is there a way to control this?

I prefer a clear upper / lower bound to a close fit. How can I tune the rules?

How can I design my second experiment to get better results?
More Questions?
Top = 1 - u
Bottom = .55 - u/2

Describe the `gap' where:

\( \text{prob}(x) < \varepsilon(u,n) \)
Unusual Functions

FF n=100000 u=.8

lower bound

ESL

1000
2000
3000
4000
5000
6000
7000
SYMBOL FONT

αβχδεφγηφικλμνοπρστυωξηζ
1234567890—[]=∶;⊂,./
!≈#∃%⊥&*()_+{}|∀<>?
ΑΒΧΔΕΦΓΗΙΘΚΛΜΝΟΠΘΡΣΤΥζΩΞΨΖ
Theory and Practice