Studies in Software Cost Model Behavior: Do We Really Understand Cost Model Performance?

Jairus Hihn

Jet Propulsion Laboratory/

California Institute of Technology

20th International Forum on COCOMO and Software Cost Modeling
October 25 to 28, 2005

BACKGROUND (1)

- Presentation is summary of a series of papers that have been recently published on the use of data mining techniques to analyze cost estimation models and data
  - › Validation methods for calibrating software effort models,
    ICSE 2005 Proceedings, May2005, St Louis, MS.
    http://menzies.us/pdf/04coconut.pdf
  - › Feature Subset Selection Can Improve Software Cost Estimation,
    PROMISE 05, May 15 2005, St Louis, MS.
    http://menzies.us/pdf/05fsscocomo.pdf
  - › Simple Software Cost Analysis: Safe or Unsafe?,
    PROMISE 05, May 15 2005, St Louis, MS.
    http://menzies.us/pdf/05safewhen.pdf
- Some of the research described was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Partial funding has been provided by NASA's Office of Safety and Mission Assurance

BACKGROUND (2)

- The common thread in these papers has been Tim Menzies currently from Portland State University
- Co-authors are various combinations of
  - **Zhihao Chen,** *University of Southern California*
  - **Tim Menzies,** *Portland State University*
  - **Daniel Port,** *University of Hawaii*
  - **Barry Boehm,** *University of Southern California*
  - **Sherry Stukes,** *Jet Propulsion Laboratory*
  - **Jairus Hihn,** *Jet Propulsion Laboratory*
- All results in this presentation are based on analysis of COCOMO 81 data sets but the techniques can be applied and basic results generalized to any model.

Open Source Data and Tools

- PROMISE repository of software engineering data sets
- COCOMO 81 (If too lazy to type it in):
  - http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff
- COCOMO 81 NASA60:
  - http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_v1.arff
  - Ground mission support software from 70's to mid-80's
- Forthcoming
  - Add historical NASA flight records from 70's to mid-80's
  - COCONUT on-line
  - Feature Subset Selection Tool
    - Google for WEKA to obtain original research software

Introduction

- The life of a Cost model
  - Typically we get some local data, run log linear regressions, and throw out the data that our engineering judgment says is an outlier
  - For models that are maintained long term, then as new projects complete we
    - Add new records - Sometimes delete old records
    - Add/modify cost drivers as new situations arise with result that models tend to grow in complexity (e.g. COCOMO II has more inputs then COCOMO81)
- But what is the basis on which we make these decisions?
  - If the R^2, F-test, and t-tests look good we call it done
  - In most cases we never formally validate.
- For small data sets with noisy data this is not good enough with the result that we <u>do not understand</u> the actual performance of our models.

Key Issues in Model Development

- What is you real estimation uncertainty?
- How many records required to calibrate?
  - Answers have varied from 10-20 just for intercept and slope
  - If we do not have enough data what is the impact on model uncertainty
- Data is expensive to collect and maintain so want to keep cost drivers and effort multipliers as few as possible
  - But what are the right ones?
  - When should we build domain specific models?
- We have had to do this piece meal if it has been done at all.
- We need to fully understand the interrelationships between all of the cost metrics we use.

Methodology

- Objective is to define a repeatable methodology that will produce better models
- Perform exhaustive search over all parameters and records in order to guide data pruning
- Measure model performance by Pred(30)
  - Number of actuals within +/- 30% of model estimate
  - Focus on mean and variance in assessing performance
  - Variance is indicator of stability and model error
    - Cost models frequently have stability problems
  - Variance computed from parameter values and model performance across multiple derived models and performance against hold out data not standard regression computations. This yields different answers.
- Calibration vs Validation Data
  - Records used to calibrate (train) and different records held out to test performance based repeated randomly selected combinations
  - {Repeats = 30, |Train| = 4, |Test| = 56, = 4}|

COCONUT:
Validation method for calibrating software effort models

- COCONUT= COCOmo, Not Unless Tuned: a baseline calibration method

- We all (correctly) believe that local calibrations improve model performance

- COCONUT can generate models with
  - Same or higher PREDs
  - Same or lower variances
  - Better extrapolation from old to new projects

COCONUT

(effort = $a*sloc^b * em_1 * em_2 *\ldots$)?

- **Assuming effort multipliers constant**
- **For i=1 to number of projects**
  - Train on 1 to i
  - Test on i+1 to N
- **For a train set,**
  - For all values of <a,b>
  - Find a' b' that minimizes error
- **For a different test set,**
  - Estimate using a' b'
  - Return PRED(20), PRED(30)
    - percentage of projects that estimate within 20/30% of actual
- **Repeat the above 30 times**
  - Randomizing order of projects, each time
  - Return mean and sd at each " i" value

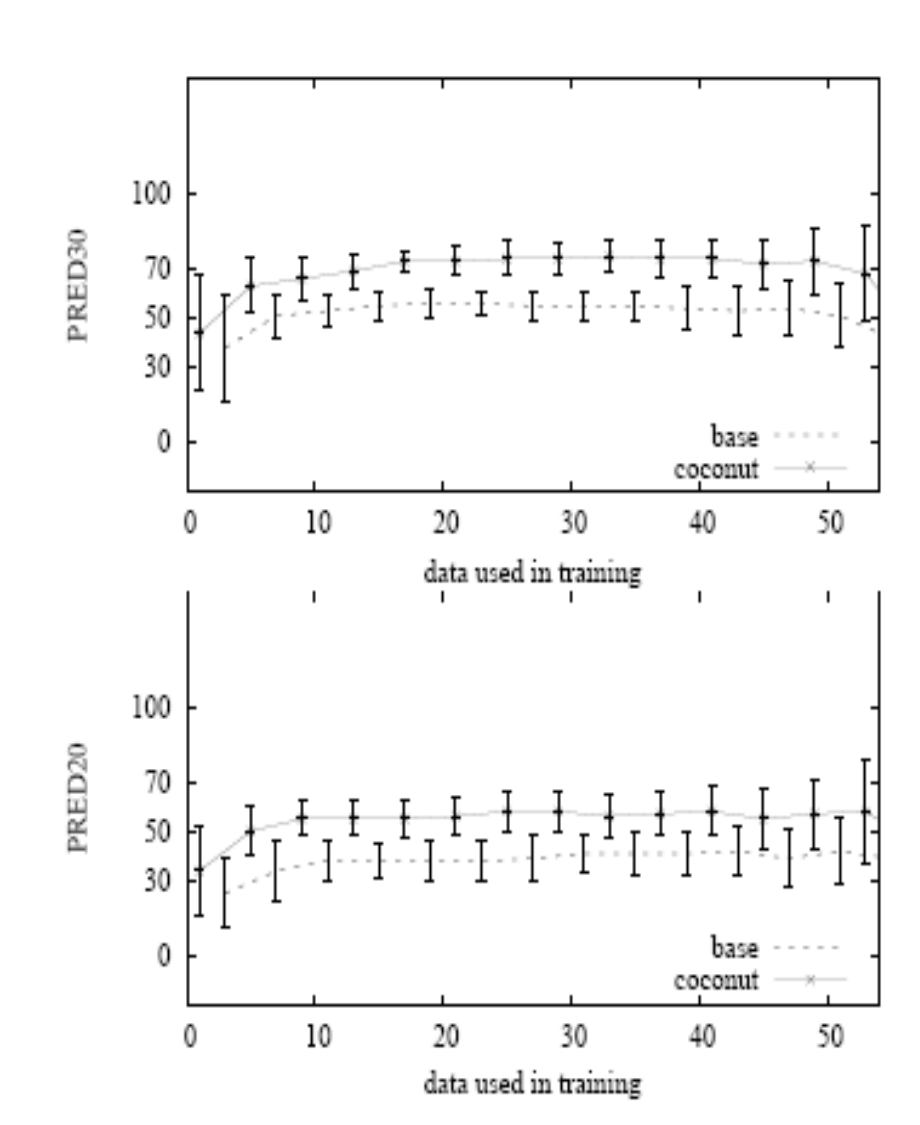
```
function train() {
    least=10**32;
    for(a=2; a<=5; a += 0.2) {
      for(b=0.9; b<=1.2; b += 0.02) {
          close =use(a,b,pred);
          if (close < least) {
          least=close;
          a'=a;
                    b'=b   }}}
    return <a',b'>}
```

COCONUT Results

base = $a*sloc^b$

cocomo81 = $a*sloc^b * em_1 * em_2 *...$

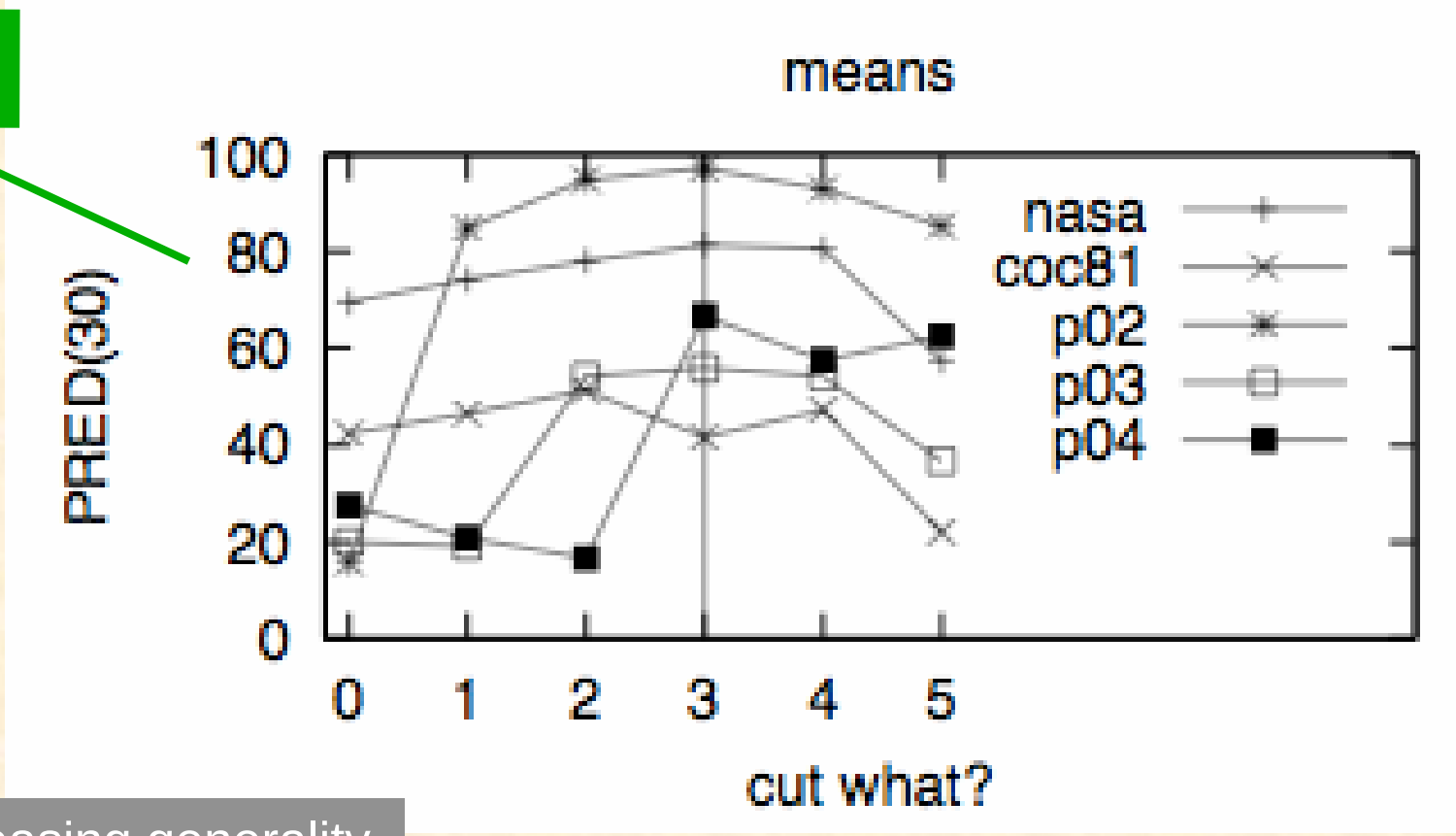
- 30 repeats (randomizing the order)
- Use t-tests to compare
  - PRED(N) using coc81 or base
  - PRED(N) after N1 or N2 projects
- Significant changes up to
  - 18 projects for PRED(30)
  - 30 projects for PRED(20)



Feature Subset Selection

Can Improve Software Cost Estimation

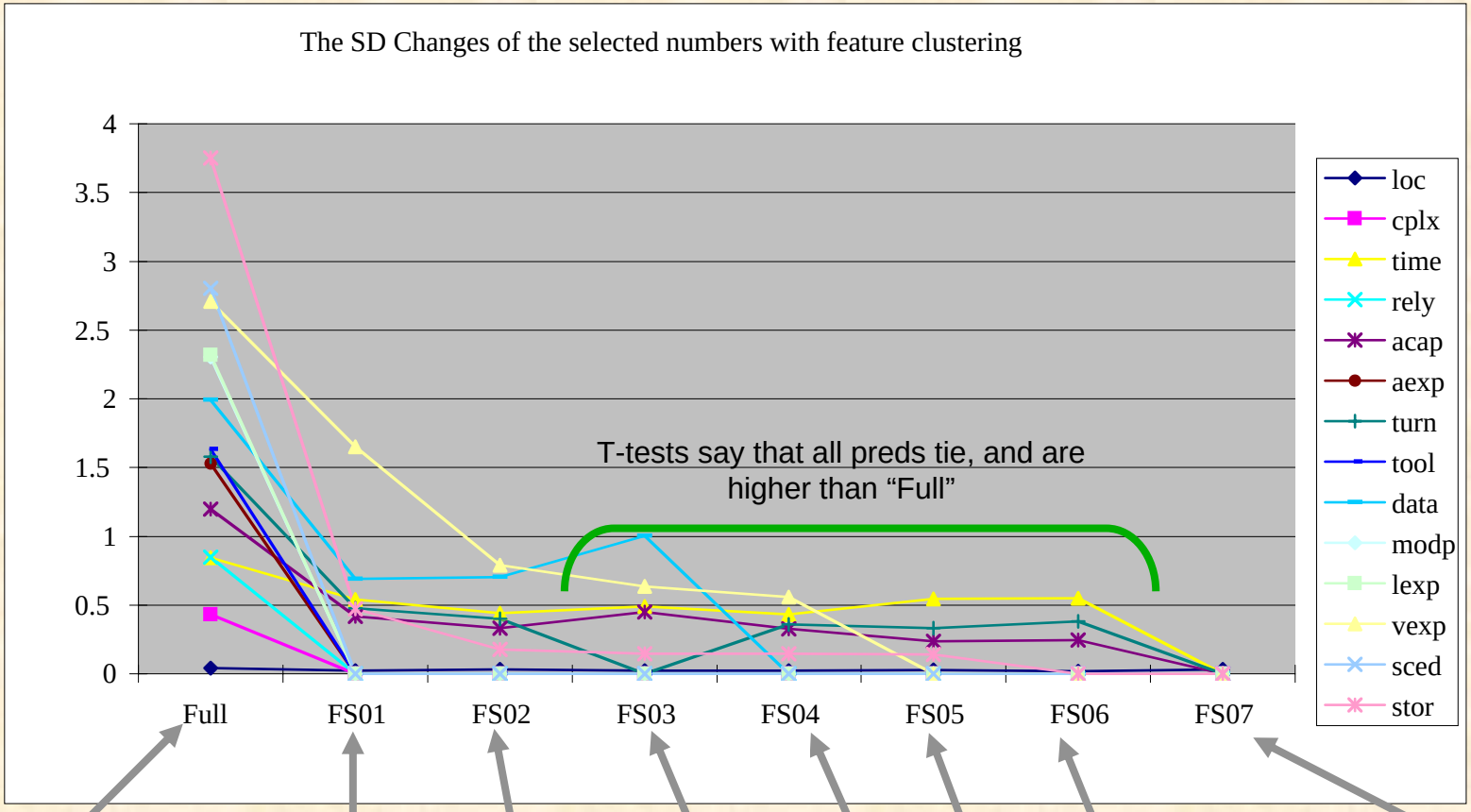
better predictions



Increasing generality (less attributes)

Feature Subset Selection

Smaller models are more stable



all

rely, lexp, tool,cplx,

aexp

data

vexp

stor

turn, time,acap

just loc

Sets defined by forward select

Feature Subset Selection

Can produce different results

- Empirically, smaller cost models have better performance
  - Always, higher pred
  - Often, less variance
- But different stratifications yield different models (cost drivers)

		loc	acap	lexp	modp	pcap	sced	time	turn	virt	
	cocomo81	✔				✔	✔	✔		✔	$\frac{5}{17}$
nasa	all	✔	✔					✔	✔		$\frac{4}{17}$
	project2	✔		✔					✔		$\frac{3}{17}$
	project3	✔			✔						$\frac{2}{17}$
	project4	✔			✔					✔	$\frac{3}{17}$
		5	1	1	2	1	1	2	2	2	

Simple Software Cost Analysis: Safe or Unsafe?

- New project cost = *delta* * (last project cost)
- *Delta* comes from COCOMO effort multipliers
  - E.g. last project: acap = v .high and rely=high
  - New project: acap = nominal, rely=low
  - New = old * (1/0.71 * 0.88/1.15 = 108%)
- Assumes "new" can be safely extrapolated from old
  - Is this always true?

	very low	low	nominal	high	very high	extra high
acap	1.46	1.19	1.00	0.86	0.71	
pcap	1.42.	1.17	1.00	0.86	0.70	
aexp	1.29	1.13	1.00	0.91	0.82	
modp	1.24.	1.10	1.00	0.91	0.82	
tool	1.24	1.10	1.00	0.91	0.83	
vexp	1.21	1.10	1.00	0.90		
lexp	1.14	1.07	1.00	0.95		
sced	1.23	1.08	1.00	1.04	1.10	
data		0.94	1.00	1.08	1.16	
turn		0.87	1.00	1.07	1.15	
virt		0.87	1.00	1.15	1.30	
stor			1.00	1.06	1.21	1.56
time			1.00	1.11	1.30	1.66
rely	0.75	0.88	1.00	1.15	1.40	
cplx	0.70	0.85	1.00	1.15	1.30	1.65

Extrapolation is safe only on some attributes

Sub-sampling experiments:
Learn models from N * 90% samples
Some attributes (e.g. X1) have unstable coefficients
Some attributes (e.g. X2) only used sometimes

	$\beta_0 +$	$\beta_1 X_1 +$	$\beta_2 X_2 +$	$\beta_3 X_3 +$	$\beta_4 X_4$
sub sample 1 :	23 +	$101X_1$ +	$21X_2$ +	$31X_3$ +	$41X_4$
sub sample 2 :	25 +	$11X_1$ +		$+ 30X_3$ +	$42X_4$
sub sample 3 :	24 +	$1X_1$ +		$+ 32X_3$	

3 * 90% samples

30 * 90% samples

9 attributes

COCOMO 81	n	β_{mean}	β_{sd}
loc	30	1.2	0.1
cplx	24	1.4	0.3
time	16	1.9	0.4
pcap	29	1.7	0.5
acap	29	2.1	0.5
rely	30	2.0	0.6
sced	25	2.9	0.7
virt	22	2.3	0.8
vexp	24	3.0	1.2

10 attributes

NASA 60	n	β_{mean}	β_{sd}	
loc	30	1.1	0.0	✔
stor	22	-0.8	0.8	✔
time	29	2.5	0.8	✔
cplx	16	1.7	1.0	✔
acap	28	2.7	1.0	✔
data	26	3.1	1.2	✔
turn	17	1.8	1.5	✘
modp	16	-1.8	1.6	✘
vexp	26	-5.6	2.3	✘
lexp	16	2.5	3.0	✘

- Only use <u>some</u> attributes can extrapolate from old to new projects
  - Many attributes missing in the sub-samples
  - Many attributes have wildly varying effects in different sub-samples

NEXT STEPS

- Infuse the described methodology into JPL and NASA
  - Develop cost model for NASA IV&V center
    - They want independent estimates of development costs to properly set their IV&V budgets
  - Locally calibrating and validating COCOMO II and SEER-SEM to JPL data
  - Apply techniques to non-software cost models
- Continue publishing and presenting results, forthcoming:
  - Specialization and Extrapolation of Software Cost Models, Proceeding in Automation in Software Engineering Conference, Menzies, Chen, Port, Hihn
  - Finding the Right Data for Software Cost Modeling, IEEE Software, Chen, Menzies, Port, Boehm
  - XOMO, 20th International Forum on COCOMO and Software Cost Modeling, Menzies
  - Many more ideas