A Next Generation Software Cost Model

Jairus Hihn
Jet Propulsion Laboratory, California Institute of Technology

Tim Menzies
Naveen Lekkalapudi
West Virginia University

James Johnson
National Aeronautics and Space Administration

June 10-13, 2014

2014 ICEAA Professional Development & Training Workshop

Introduction

- Purpose of this talk is to describe a new clustering algorithm that can be used to estimate software size and effort that is effective for
  - small sample sizes
  - noisy data
  - and uses high level systems information

Background

- The *NASA Software CER Development Task is funded by the Cost Analysis Division to develop a software cost model that*
  - *Can be used in the early lifecycle*
  - *Can be used effectively by non-software specialists*
  - *Uses data from NASA in-house built and funded software "projects"*
    - *CADRe but also other Center level data sources*
  - *Supplement to current modeling and bottom up methods not a replacement*
  - *Can be documented as a paper model*
  - *Acceptable for use with both the cost and software communities*
- *Year 1 building a prototype model for robotic flight software*

Data Sources

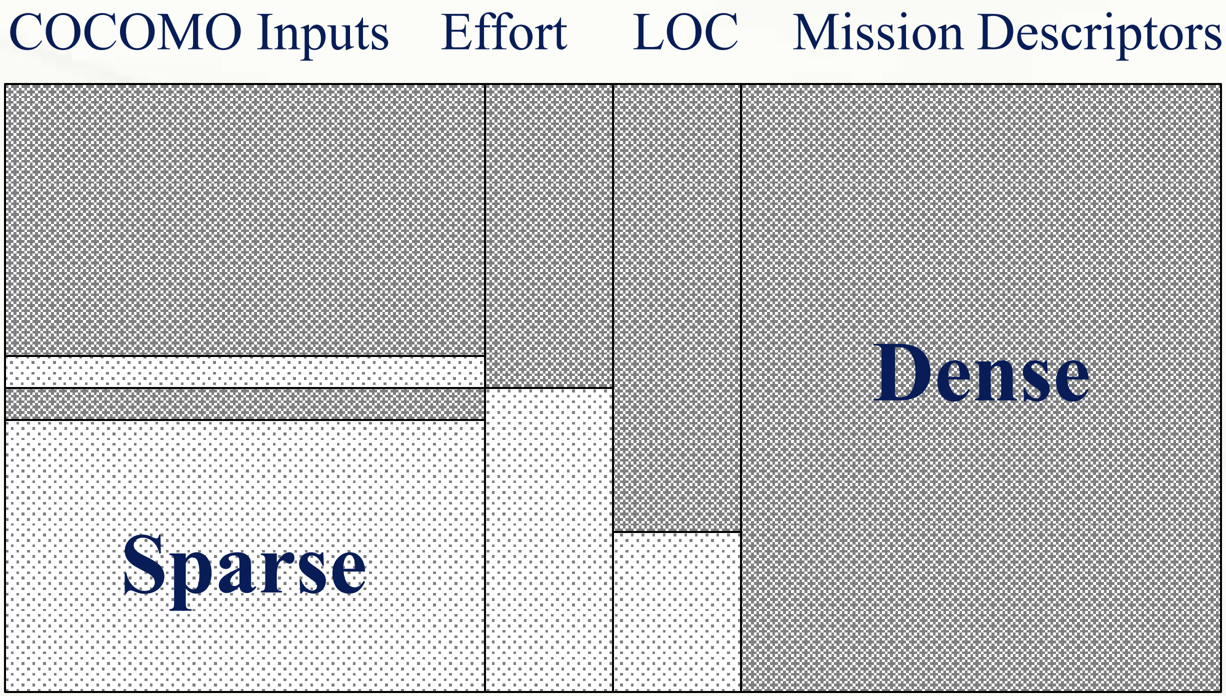
- Where the data came from
  - CADRe
  - NASA 93 - Historical NASA data originally collected for ISS (1985-1990) and extended for NASA IV&V (2004-2007)
  - Contributed Center level data
  - NASA software inventory
  - Project websites and other sources for system level information if not available in CADRe

Data Items

- Total development effort in work months
- Delivered and equivalent logical lines
- COCOMO model inputs
  - Translated from CADRE which has SEER model inputs
- System parameters
  - Mission Type (deep-space, earth-moon, rover-lander, observatory)
  - Multiple element (probe, etc.)
  - Number of instruments (Simple, Medium&Complex)
  - Number of deployables (Simple, Medium&Complex)
  - Flight Computer Redundency
  - Heritage

Data Yield

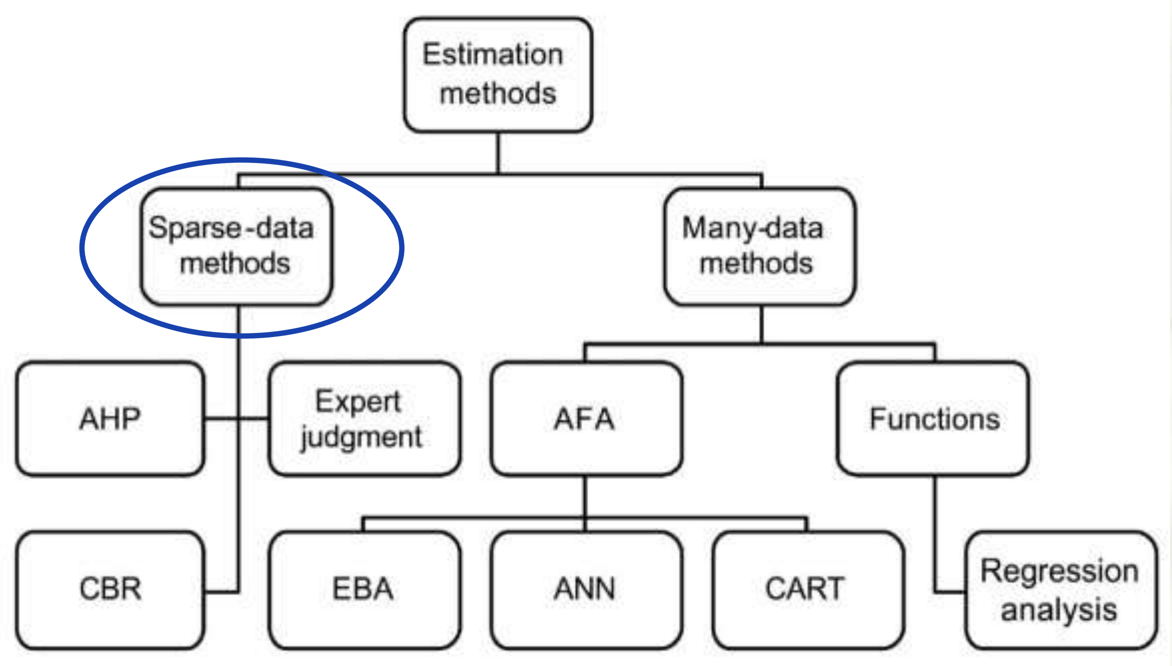
- 39 records with system descriptors mostly from GSFC and JPL
- 19 records have all data items
- 31 records have delivered LOC
- 21 records have effort



Why explore alternative modeling methods?

Because different methods exist for a reason

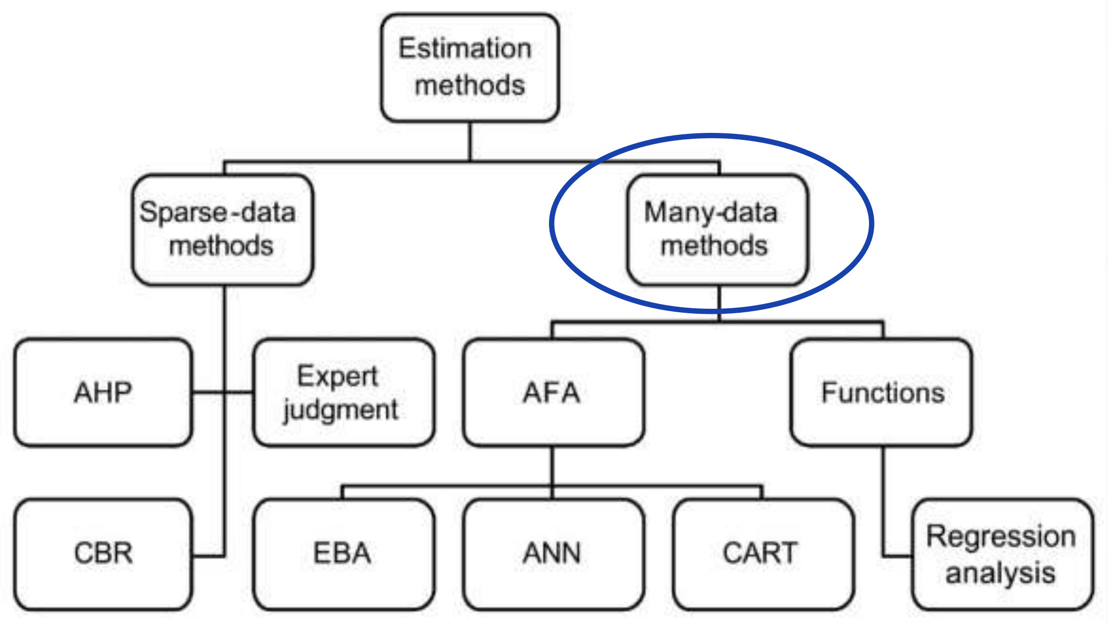
Effort Estimation Methods



Sparse-data methods:

- **Analytic Hierarchy Process (AHP)**
  - Find concurrent solutions to sub-problems
- **Expert Judgment**
  - Use expert's estimation knowledge
  - Jorgensen's 12 best practices
- **Automated Case-Based Reasoning (CBR)**
  - Find similarities between past projects' solutions (cases) and the current one

Many-data Estimation Methods



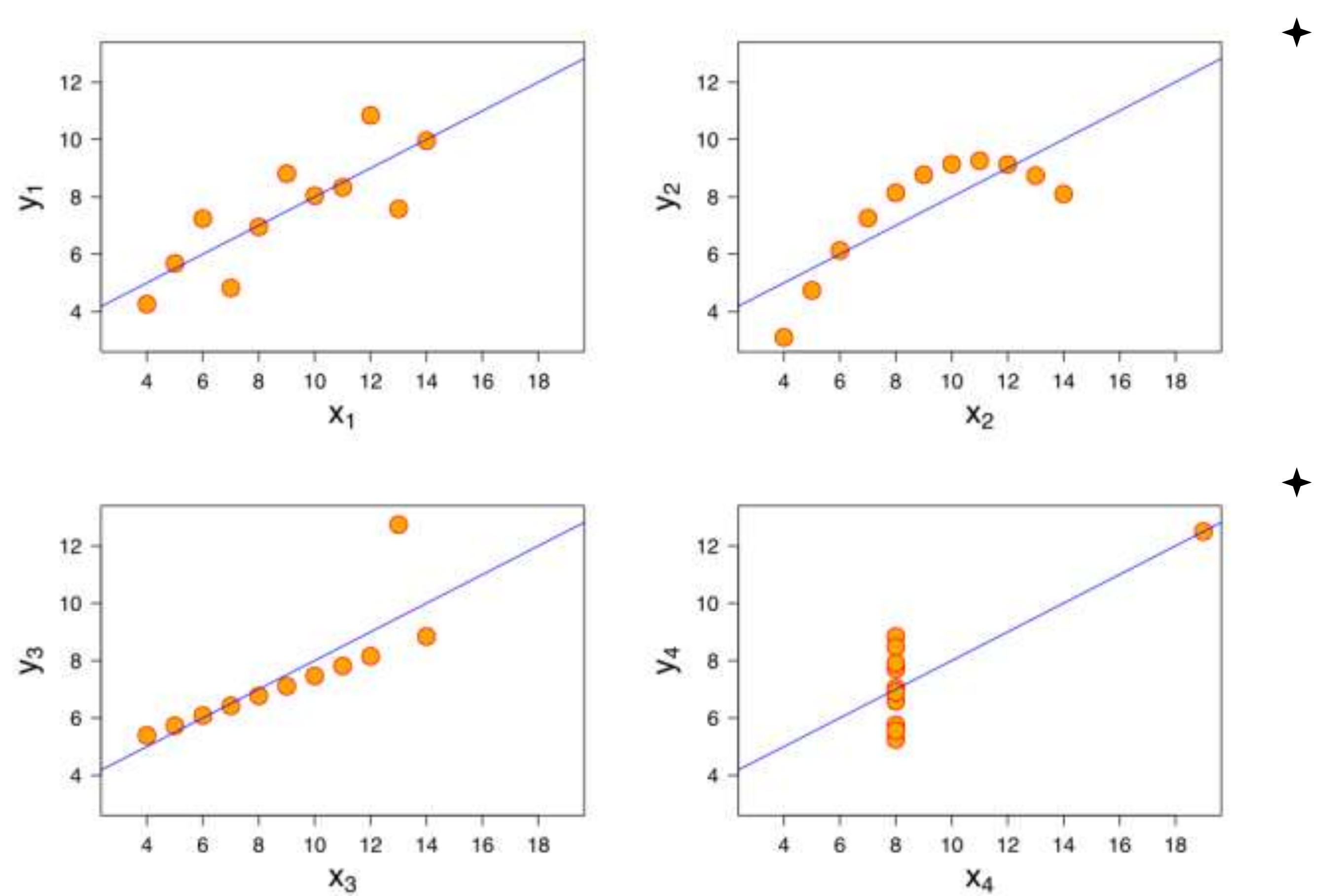
Many-data methods:

- Functions: mathematical relation between variables ($y=ax^b$)
  - Regression Analysis
- Arbitrary Function Approximators (AFA): no such relation between x and y
  - Estimation by Analogy (EBA): nearest neighbor
  - Artificial Neural Networks (ANN)
  - Classification and Regression Trees (CART)

Anscombe's Quartet

Models especially regression models built on small samples with noisy data can be very misleading

Anscombe's Quartet

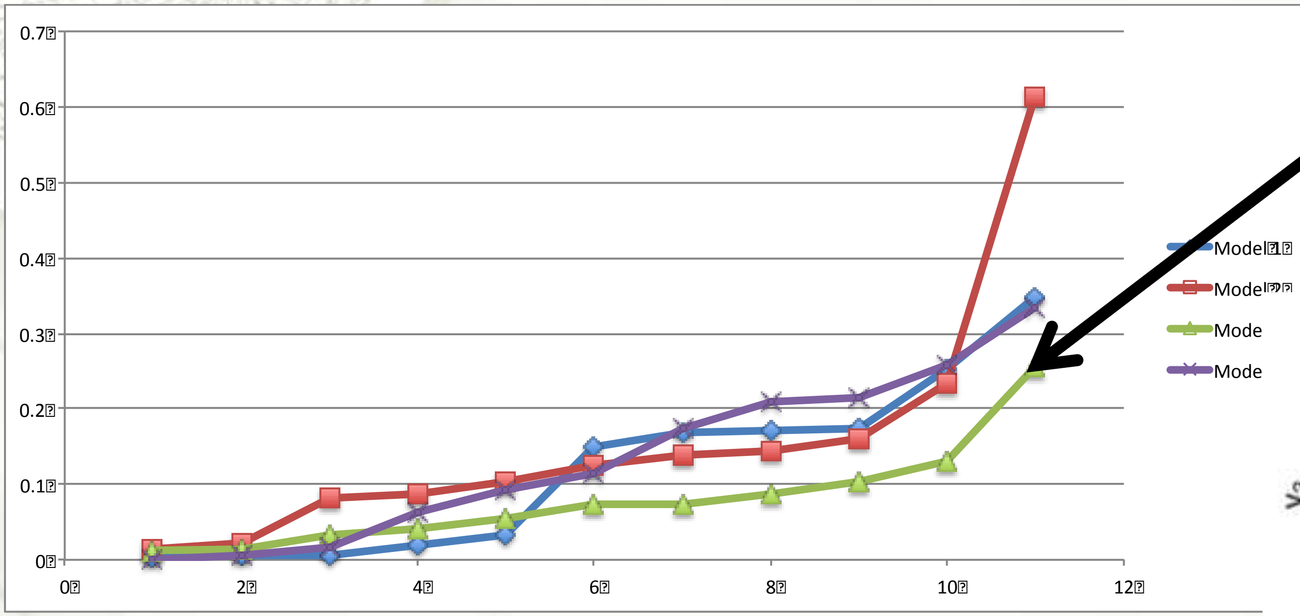


- All four of the displayed plots have virtually identical statistics
  - Means, Medians, Variances
  - Regression line, R^2, F and T tests
- But visual inspection clearly shows they are very different

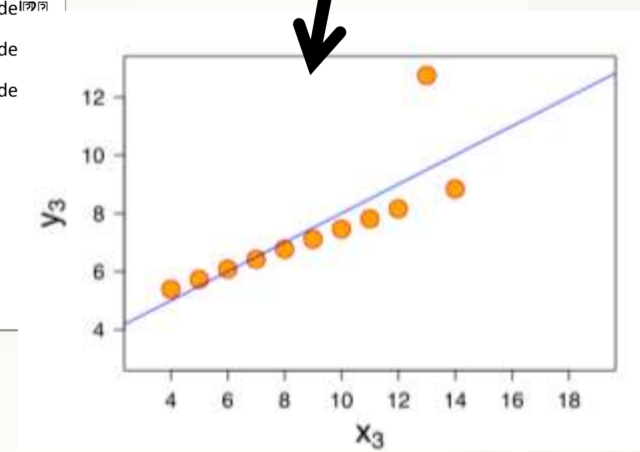
Reference: Anscombe, F. J. (1973). "Graphs in Statistical Analysis". *American Statistician* **27** (1): 17–21. JSTOR 2682899. Can also be found at http://en.wikipedia.org/wiki/Anscombe%27s_quartet

Anscombe's Quartet – Using MRE

- MRE can distinguish between the models



Model 3 fits its data the best



- Plotting the absolute values of the relative error it is easily seen that Model 3 fits its data best just as intuition would indicate
  - MRE = Magnitude of Relative Error, abs(Predicted – Actual)/Actual

Data Mining Methods

- Data mining techniques provided us with the rigorous tool set we needed to explore the many dimension of the problem we were addressing in a repeatable manner
  - Analyze standard and non-standard models
    - Is there a best functional form
  - Perform exhaustive searches over all parameters and records in order to guide data pruning
    - Rows (Stratification)
    - Columns (variable reduction)
  - Measure model performance by multiple measures
    - R^2, MRE, Pred, F-test, etc.
  - Is there a ‘best’ way to tune or calibrate a model
  - How important is it to us different calibration and validation datasets

Effort Estimation with Data Mining Methods References

"Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data" by Ekrem Kocaguneli and Tim~Menzies and Jacky Keung and David Cok and Ray Madachy. IEEE Transactions on Software Engineering (pre-print) 2013 . .

"Finding conclusion stability for selecting the best effort predictor in software effort estimation" by J. Keung and E. Kocaguneli and T. Menzies. Automated Software Engineering pages 1-25 May 2012 . Available from http://menzies.us/pdf/12findstable.pdf .

"Exploiting the Essential Assumptions of Analogy-Based Effort Estimation" by E. Kocaguneli and T. Menzies and A. Bener and J. Keung. IEEE Transactions on Software Engineering pages 425-438 2012 . Available from http://menzies.us/pdf/11teak.pdf .

"Local vs. Global Lessons for Defect Prediction and Effort Estimation" by Menzies, T. and Butcher, A. and Cok, D. and Marcus, A. and Layman, L. and Shull, F. and Turhan, B. and Zimmermann, T.. IEEE Transactions on Software Engineering pages 1 2012 . Available from http://menzies.us/pdf/12localb.pdf .

"Kernel methods for software effort estimation" by E. Kocaguneli and T. Menzies and J. Keung. Empirical Software Engineering pages 1-24 2011

"On the Value of Ensemble Effort Estimation" by Kocaguneli, E. and Menzies, T. and Keung, J.. IEEE Transactions on Software Engineering 2011

"Exploring the Effort of General Software Project Activities with Data Mining" by Topi Haapio and Tim Menzies. International Journal of Software Engineering and Knowledge Engineering pages 725-753 2011

"Stable Rankings for Different Effort Models" by Tim Menzies and Omid Jalali and Jairus Hihn and Dan Baker and Karen Lum. Automated Software Engineering December 2010 . Available from http://menzies.us/pdf/10stable.pdf .

"Case-Based Reasoning for Reducing Software Development Effort" by Adam Brady and Tim Menzies and Oussama El-Rawas and Ekrem Kocaguneli and Jacky Keung. Journal of Software Engineering and Applications 2010 . Available from http://menzies.us/pdf/10w0.pdf .

"A Second Look at Faster, Better, Cheaper" by Oussama El-Rawas and Tim Menzies. Innovations Systems and Software Engineering pages 319-335 2010 . Available from http://menzies.us/pdf/10bfc.pdf .

"Explanation vs Performance in Data Mining: A Case Study with Predicting Runaway Projects" by Tim Menzies and O. Mizuno and Y. Takagi and Y. Kikuno. Journal of Software Engineering and Applications pages 221-236 November 2009

"Accurate Estimates Without Local Data?" by Tim Menzies and S. Williams and Oussama El-Rawas and D. Baker and B. Boehm and J. Hihn and K. Lum and R. Madachy. Software Process Improvement and Practice pages 213-225 July 2009 . Available from http://menzies.us/pdf/09nodata.pdf .

"Selecting Best Practices for Effort Estimation" by Menzies, Tim and Chen, Zhihao and Hihn, Jairus and Lum, Karen. IEEE Transactions on Software Engineering pages 883--895 doi = 10.1109/TSE.2006.114 issue = 11 2006

Spectral Clustering

- Find eigenvectors in data
  - Recursively splits the data on synthesized dimension of greatest variance
  - Principal Component Analysis (PCA) is also an eigenvector method
  - Spectral Clustering is like PCA on steroids
- Why use it
  - If noisy variables: they will disappear
  - If irrelevant variables: they will be ignored
  - If correlated variables: they will be combined together into an eigenvector

Estimation Experiment 1

- Given a set of mission descriptors
- How well can we estimate software system size?
  - Estimate delivered LOC range which could be used as input into COCOMO, SEER or other software cost models
  - Use spectral clustering
    - **Centroid** = use centroid of nearest cluster
      - Test whether mean, median is best
    - **Interpolation** = interpolate in between the two nearest clusters
      - Test whether mean, median is best

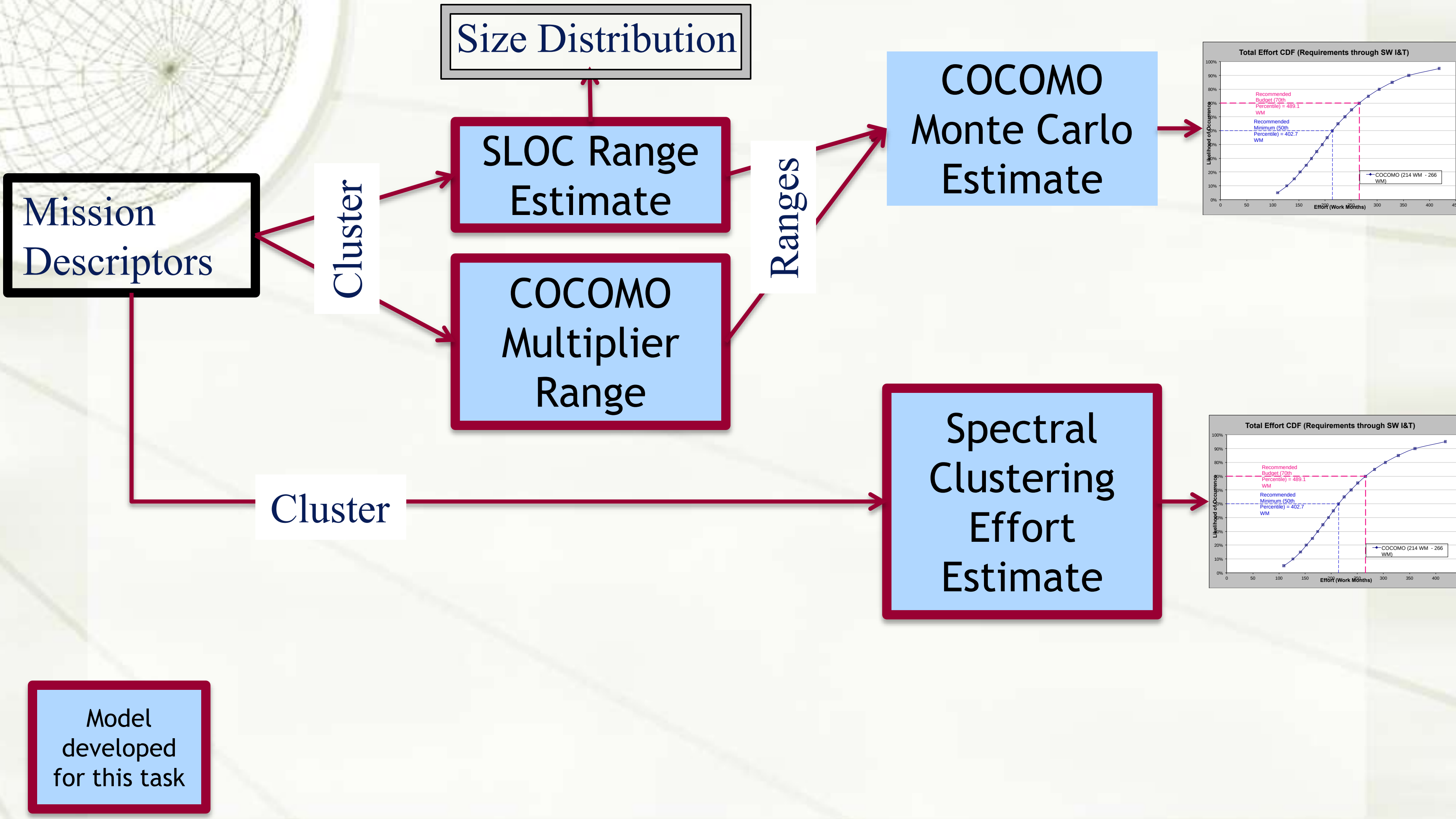
Estimation Experiment 2

- Experiment 2: Given a set of mission descriptors How well can we estimate development effort?
  - Uses spectral clustering only with system descriptors
    - **Centroid** = use centroid of nearest cluster
      - Test whether mean, median is best
    - **Interpolation** = interpolate in between the two nearest clusters
      - Test whether mean, median is best
  - Is this method as good as using a standard cost model?

Estimation Experiment 3

- Experiment 3: Given a set of mission descriptors How well can we estimate development effort with COCOMO?
  - Hold out 1 project
  - Do spectral clustering with both COCOMO inputs and System descriptors for both LOC and COCOMO Effort Multipliers
  - Find two nearest clusters and interpolate which yields a range for LOC and EM’s
  - Run COCOMO using ranges to derive an effort distribution
  - Comparing estimate to actual to evaluate

Estimation Experiments

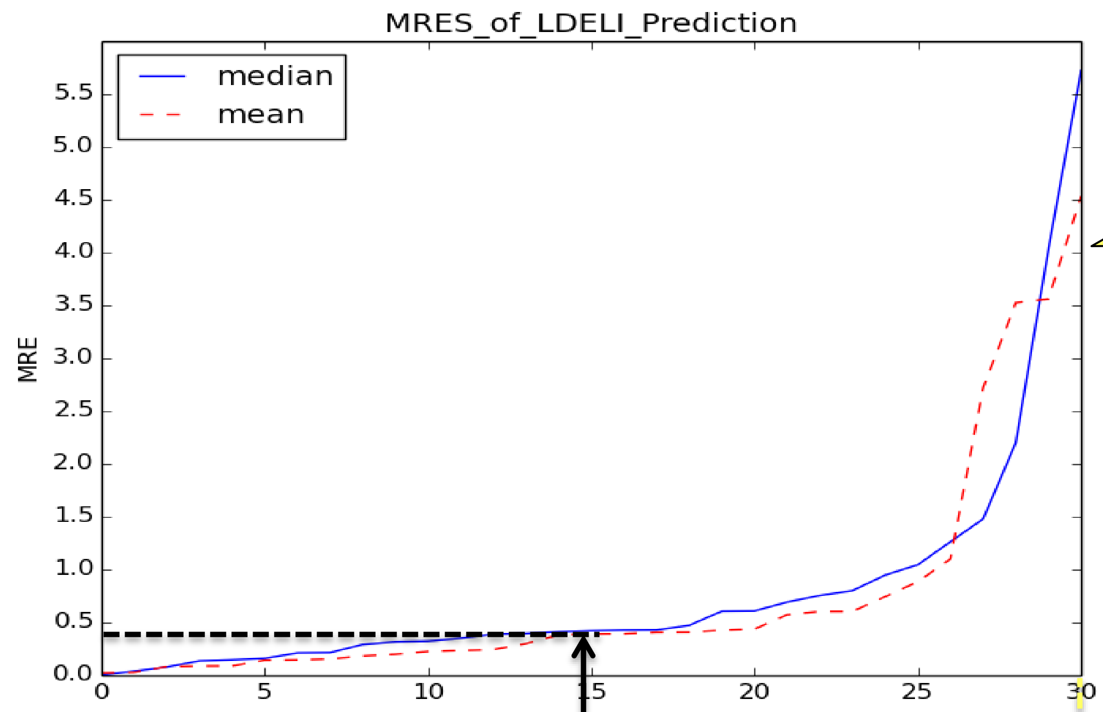


Methodology Results

- Pure clustering
  - Median measures always win
    - Has implications for our commonly used regression based models which are regression to the mean
  - Interpolation beats centroid
    - Produces lower over all MRE
  - **Median distance between two clusters is best**
    - Produces lower over all MRE

SLOC Estimation

- Results so far are promising
  - Remember that software size growth of 50-100%+ is not uncommon

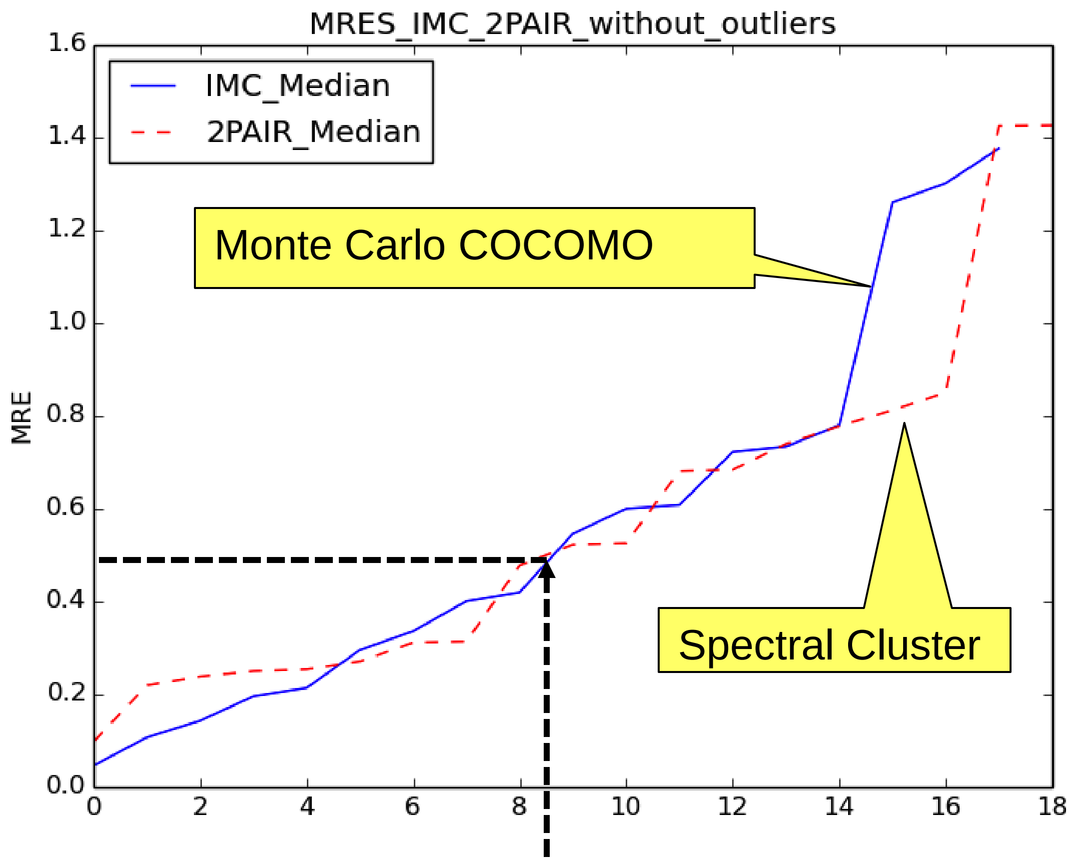


3 major outliers need to look into

Half the time, estimates within 40% of actual, using early life cycle data

Comparing Estimates: Model vs Clustering

There is no difference!



- Clustering using just high level system descriptors/variables estimates just as well as running the COCOMO model
- There is no inherent reason to assume with similar inputs that other models would perform and better

Half the time, estimates within 50% of actual, using early life cycle data

Conclusions and Next steps

- Initial results very promising:
  - Reasonably accurate LOC estimators for very early lifecycle data
  - Effort estimators for very early lifecycle data.

- Next Steps under consideration
  - Expand and improve SC flight software data set and improve results
  - Add Instrument flight software
  - Test with SEER-SEM
  - Document model
  - Further explore combinations of data sets and methods for constructing clusters
  - Engage NASA software and cost community on how to pilot and improve the models