



# **A Next Generation Software Cost Model: a look under the hood**

***Jairus Hihn***

*Jet Propulsion Laboratory, California Institute of Technology*

***Tim Menzies***

***Naveen Lekkalapudi***

*West Virginia University*

***James Johnson***

*National Aeronautics and Space Administration*

August 11-13, 2014

2014 NASA Cost Symposium



# Introduction

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ Purpose of this talk is to describe a new NASA Software Cost Model that is under development
- ★ It is built around a spectral 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

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ *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

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ Where the data came from
  - ★ CADRe (When it exists and is usable)
  - ★ 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

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ✦ 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 Redundancy
  - ✦ Heritage



# System Descriptor Details (Example)

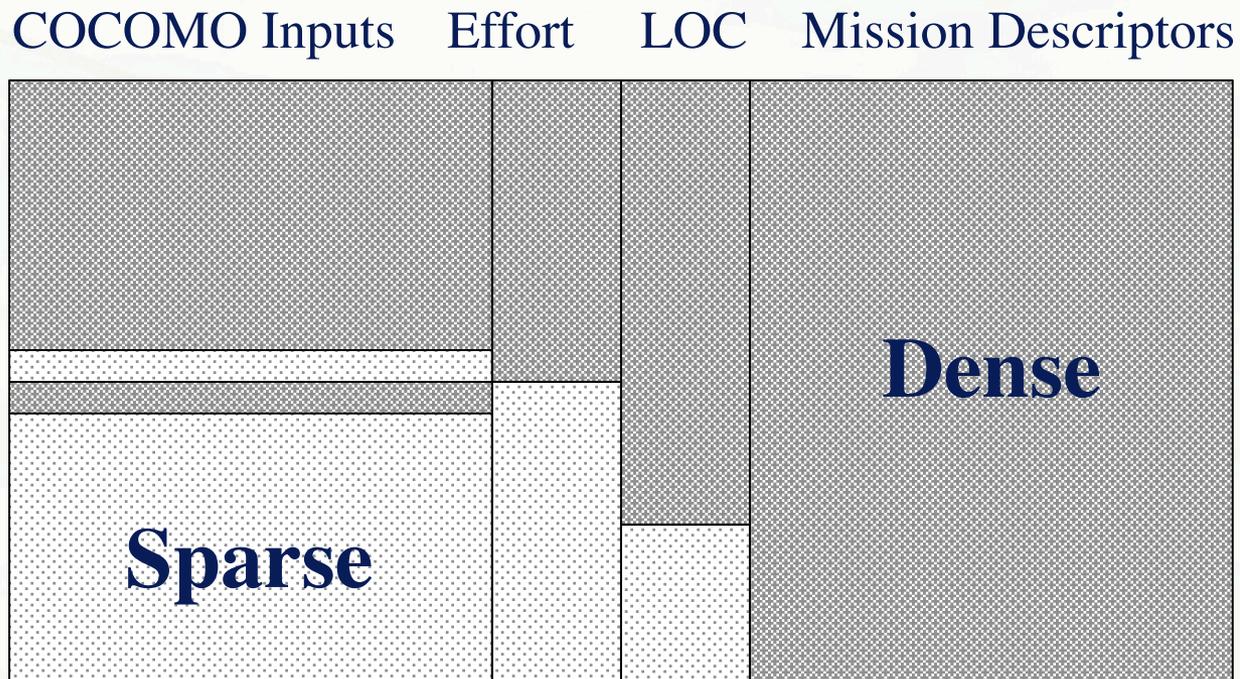
| System Descriptors |                     |   |                              |
|--------------------|---------------------|---|------------------------------|
| Mission Type       | Values              | Description   | Example                      |
|                    | Earth/Lunar Orbiter | Robotic spacecraft that orbit the earth or moon conducting science measurements. These spacecraft are very similar if not identical to the many commercial satellites used for communication as well as many military satellites. They often can have high heritage and even use production line buses from industry. | Aqua                         |
|                    | Telecomm Sat        | Earth orbiters that support very high bandwidth and designed for very long life.  | TDRS                         |
|                    | Observatory         | Observatories are space based telescopes that support space based astronomy across a wide set of frequencies. They can be earth orbiters or earth trailing at the various lagrange points created by the gravoty fields of the earth, sun and moon.   | Hubble                       |
|                    | Deep Space          | Any robotic sapcecarft that goes beyond the moons orbit. So this category includes any misison whose destination is a planet, planetoids, any planetary satellite, comet, asteroid or the sun. These misison can be orbiters or flybys or a mixture of both.  | Deep Impact                  |
|                    | Static Lander       | A robotic spacecraft that does its science in-situ or from the surface of a soplar system body. It does not move from its original location.  | Phoenix                      |
|                    | Rover               | A robotic spacecraft that does its science in-situ or from the surface of a solar system body and has the ability to move on the surface. To date all rovers have wheels but in the future they may crawl, walk or hop.   | Mars Exploration Rover (MER) |

★ Complete list is in the backup slides



# 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?*

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ For most of our history the cost community has relied upon regression type modeling methods
  - ★ Regression method have the underlying assumption of
    - ★ clean and complete data with large sample sizes
- ★ Cost data suffers from sparseness, noise, and small sample sizes
- ★ There are alternative methods that handle these conditions better then regression



## *Anscombe's Quartet*

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

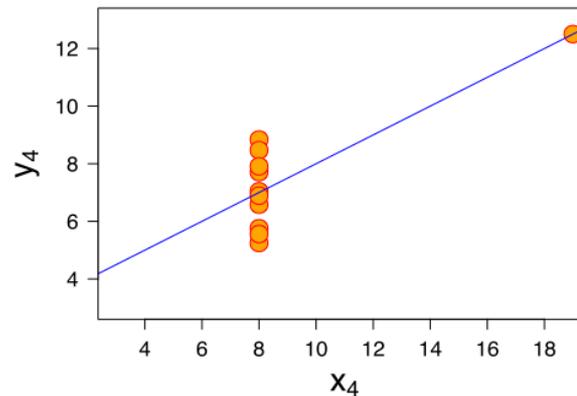
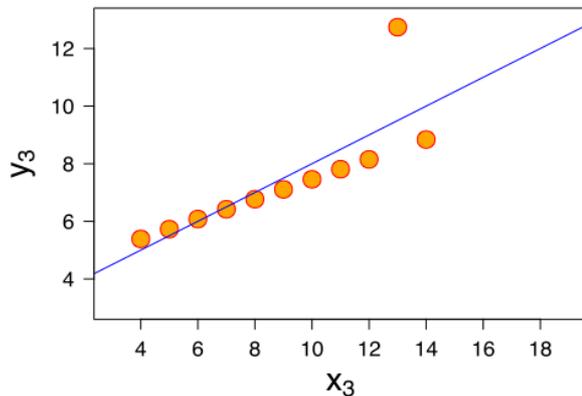
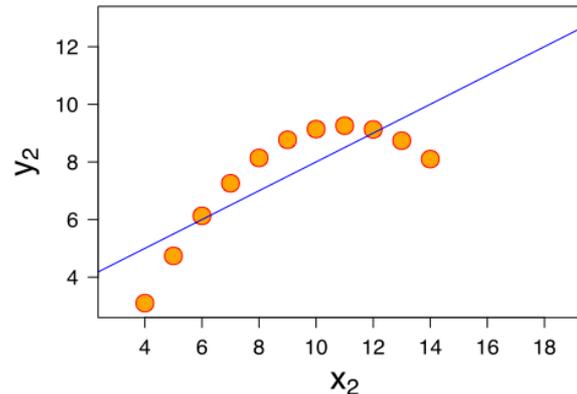
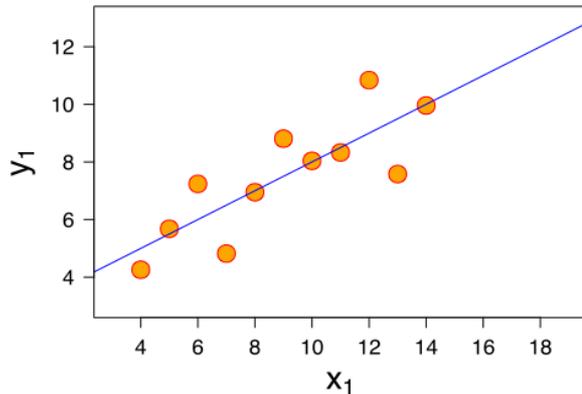


# Anscombe's Quartet

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University



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



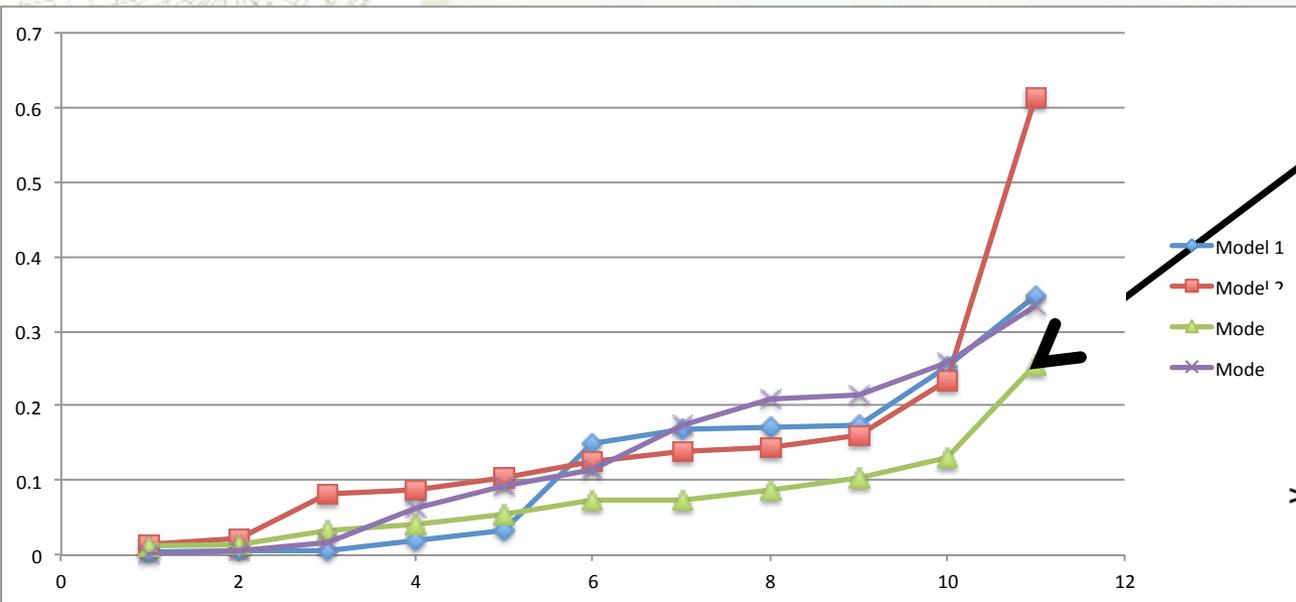
# Anscombe's Quartet - Using MRE

Cost Analysis Division

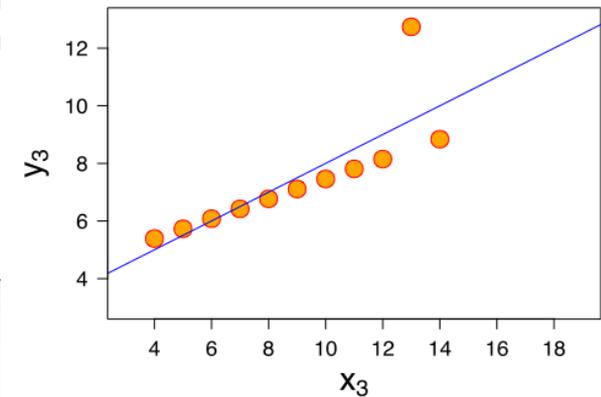
Jet Propulsion Laboratory

West Virginia University

✦ 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,  $\text{abs}(\text{Predicted} - \text{Actual})/\text{Actual}$

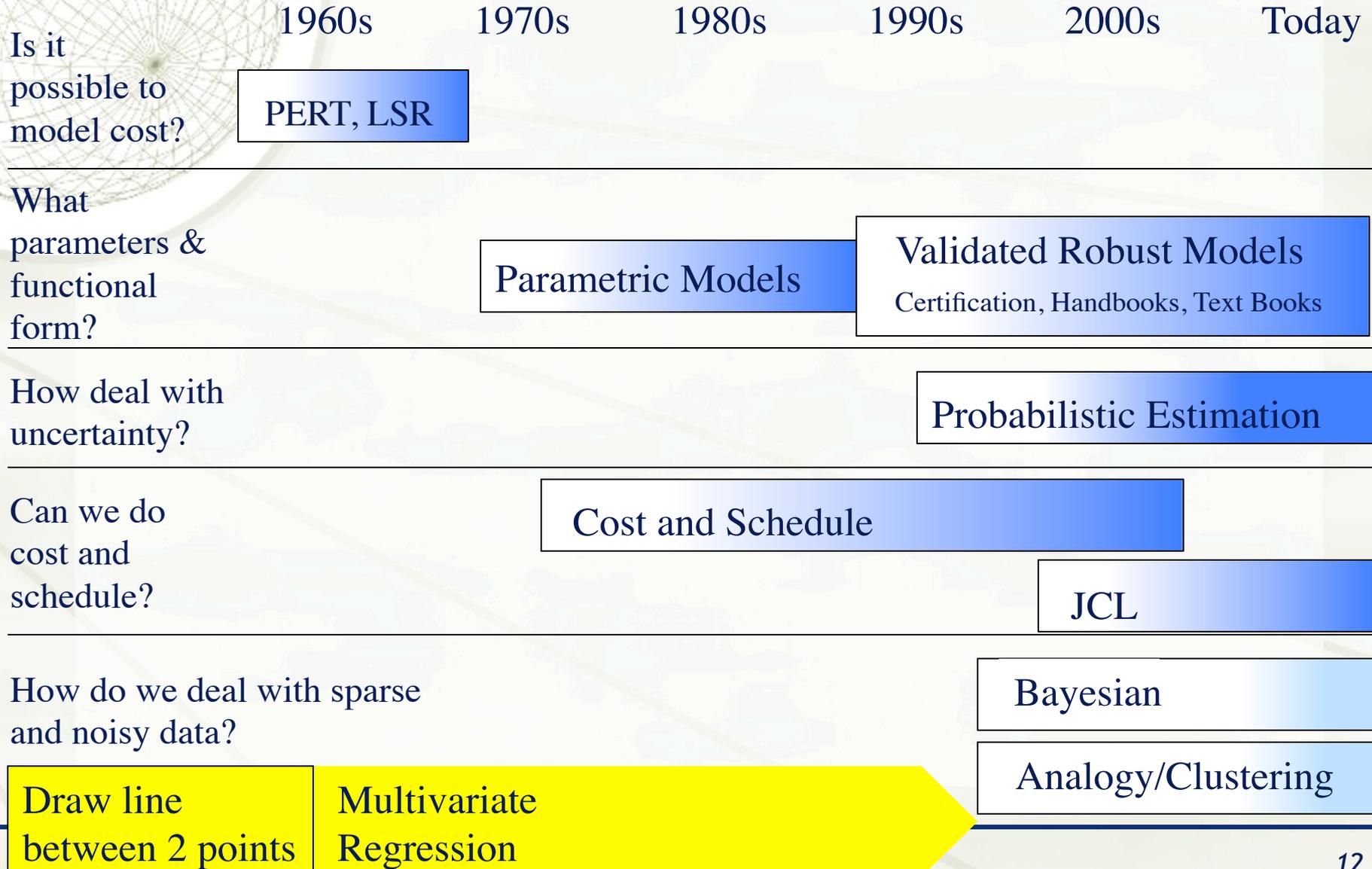


# Evolution of Methods

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University





# Data Mining Methods

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ 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



"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

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ PCA finds eigenvectors in numerical data
- ★ Spectral Clustering
  - ★ Spectral Clustering is like PCA on steroids but uses an eigenvector approximation method
  - ★ Recursively splits the data on synthesized dimension of greatest variance/spread
- ★ Why use it
  - ★ Can handle numerical and symbolic data
  - ★ Can work on small, sparse and somewhat noisy data sets but also works well on large consistent data sets
  - ★ Can use as estimator with partial information

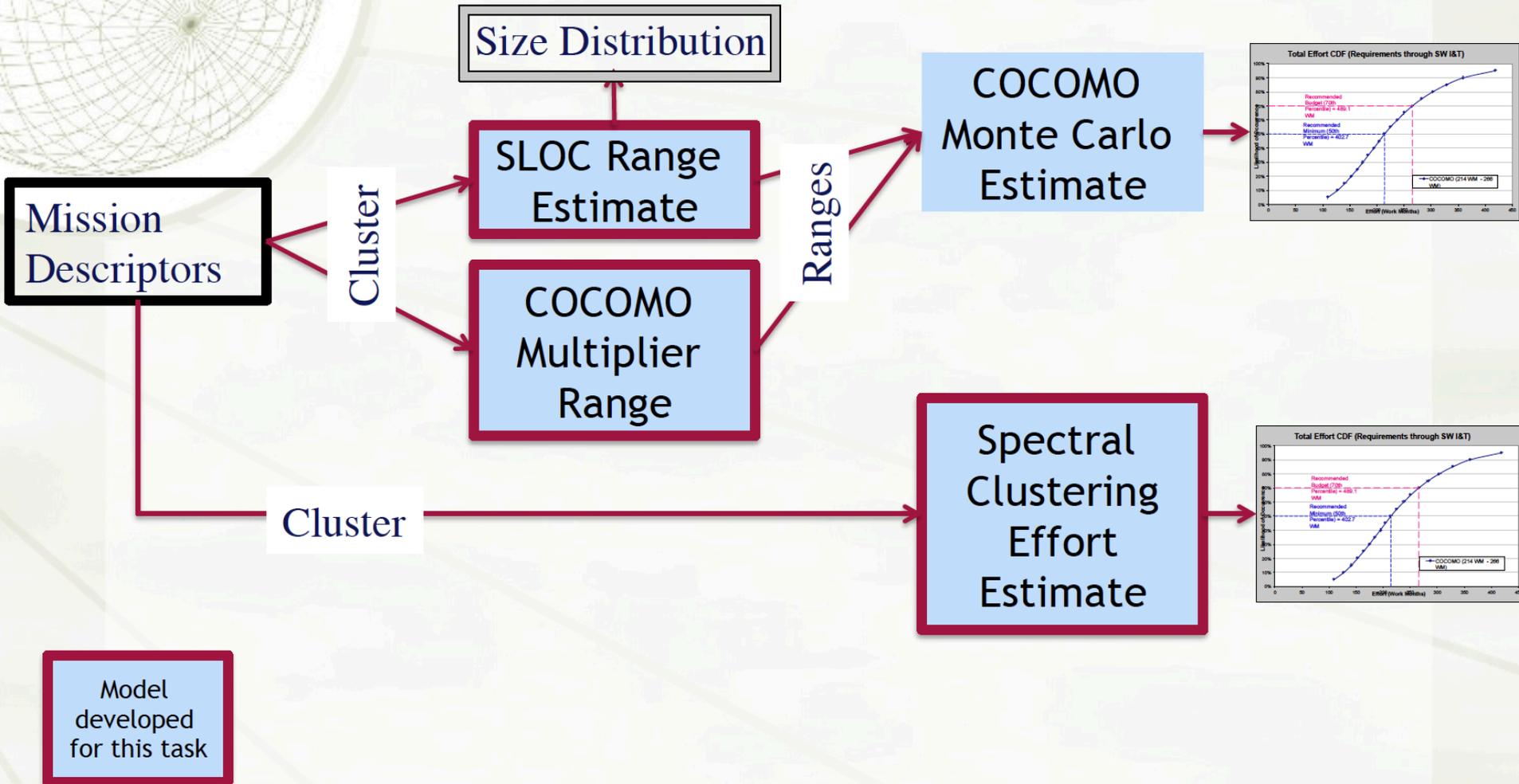


# Estimation Experiments

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University





# Side Note - Methodology Results

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ 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



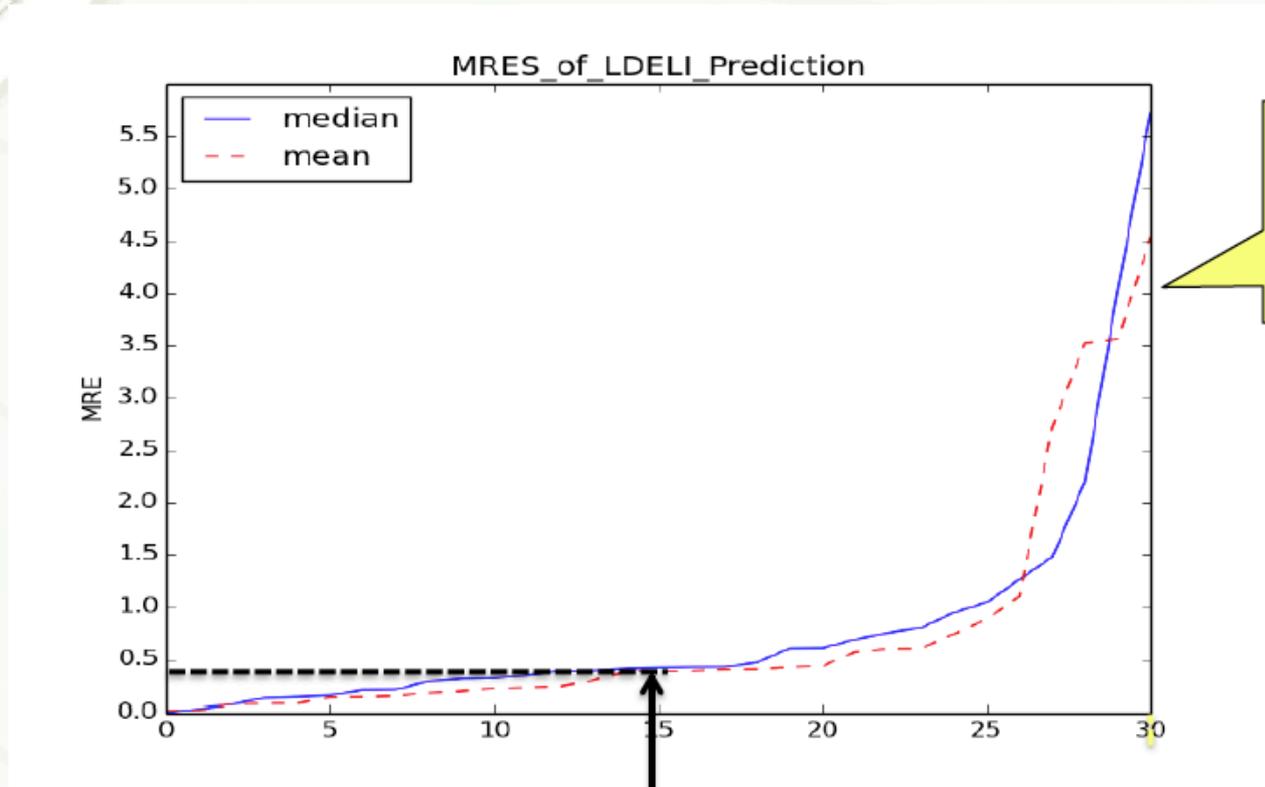
# SLOC Estimation

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

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

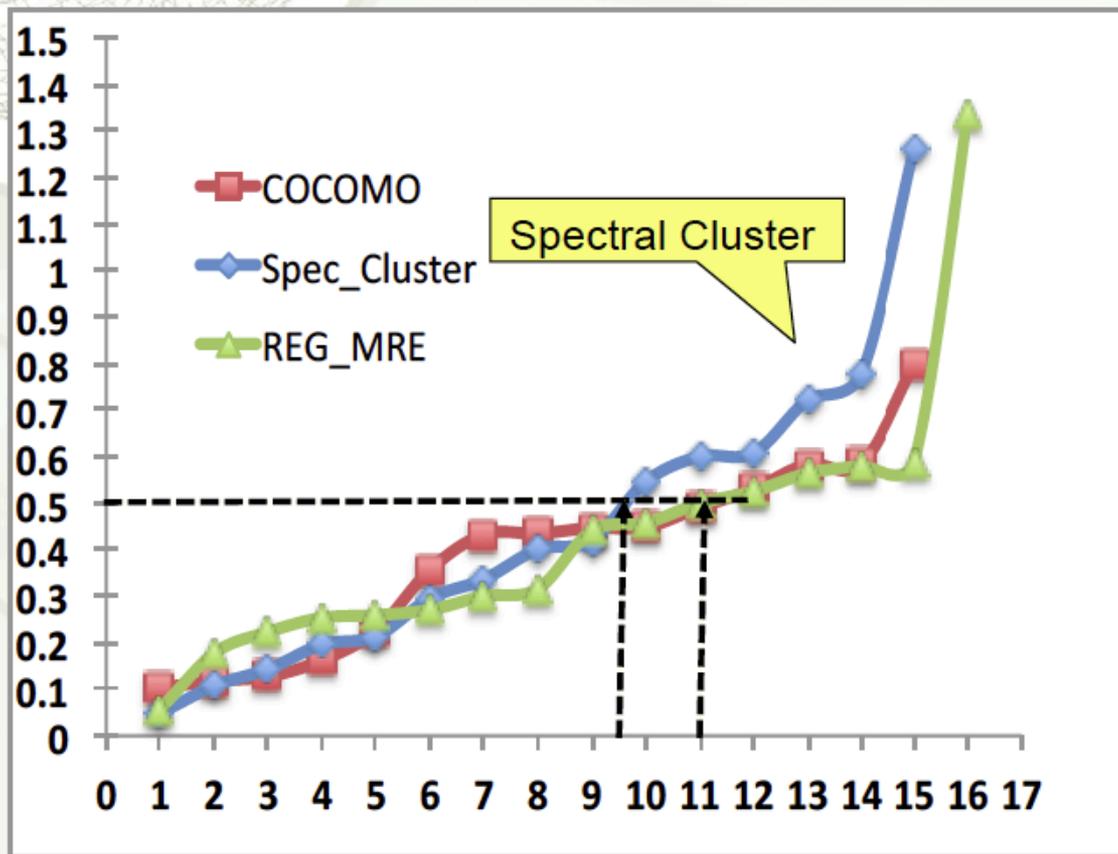


3 major outliers need to look into

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



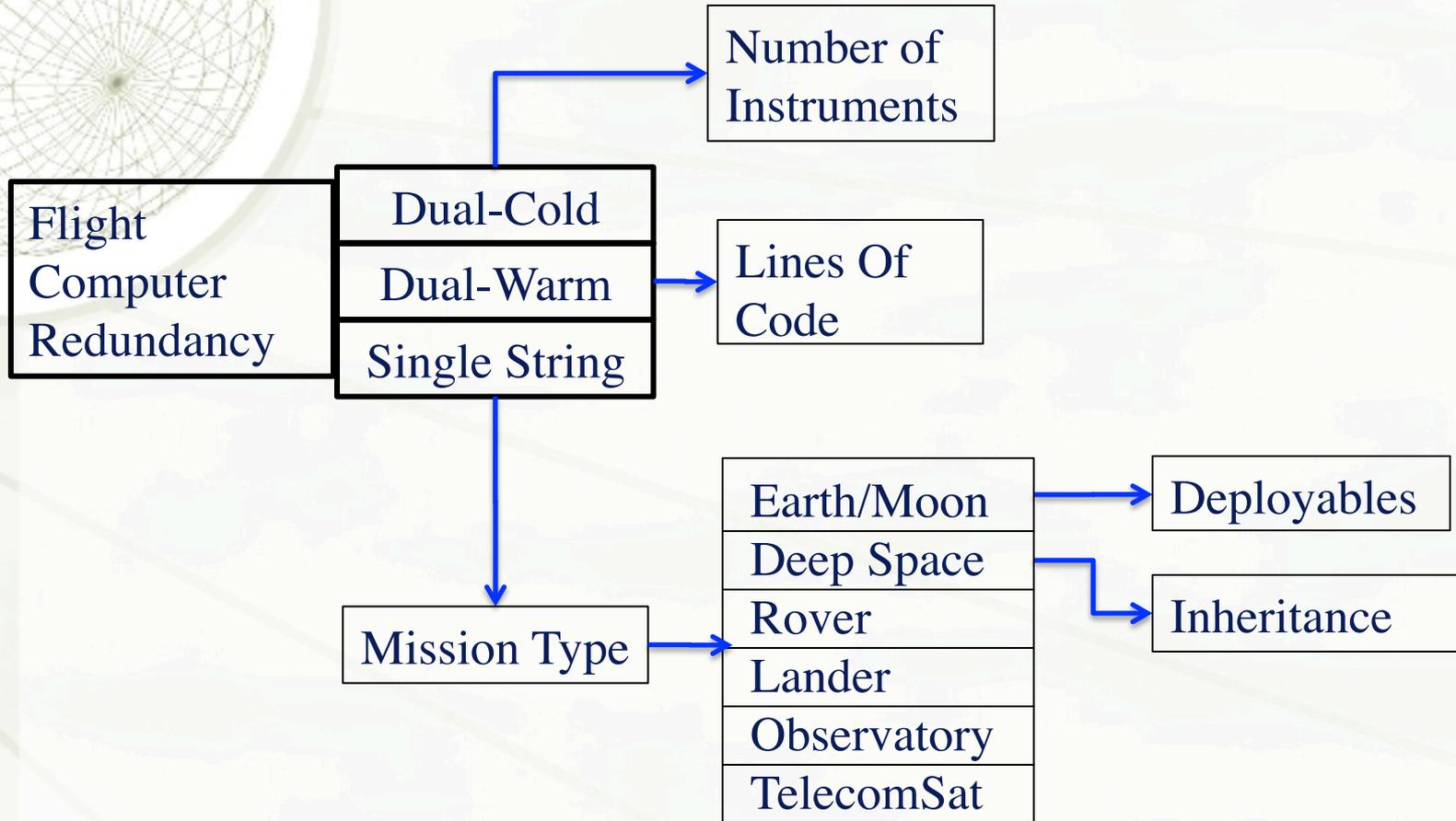
## Clustering on Systems Parameters does almost as well as COCOMO or a Regression!



- ◇ Clustering using just high level system descriptors/ variables estimates almost as good as running the COCOMO model or a simple regression
  - ◇ LSR - Effort/EM =  $aS^b$
  - ◇ Results biased
- ◇ There is no inherent reason to assume with similar inputs that other models would perform any better

67-73% of estimates within +/-50% of actual, using early life cycle data

# Cluster Discriminators





# NASA SW Cluster Estimation Prototype

## Example Clusters

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

### COCOMO EM

### Enter Data

### Rovers

### System Descriptors

| \$Id | \$StartYear | \$Sprec | \$Sflex | \$Sresl | \$Ssteam | \$Spmat | \$Srely | \$Scplx | \$Srupe | \$Stime | \$Sstor | \$Spvol | \$Sacap | \$Spcap | \$Spcan | \$Spex | \$Sfltx | \$Sstool | \$Ssced | \$Ssite | \$Sdocu | \$SlogicalDeliver | Language       | Mission Type       | Secondary Types | Complexity | \$NumSimpleInstruments | \$NumMediumInstruments | \$NumInStr | Flight Computer Redundancy | \$NumSimpleDeployables | \$NumMediumDeployables | \$NumComplexDeployables | Inheritance | Effort    |        |
|------|-------------|---------|---------|---------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|--------|---------|----------|---------|---------|---------|-------------------|----------------|--------------------|-----------------|------------|------------------------|------------------------|------------|----------------------------|------------------------|------------------------|-------------------------|-------------|-----------|--------|
| 1997 |             |         |         |         |          |         |         |         |         |         |         |         |         |         |         |        |         |          |         |         |         | C                 | DeepSpace      | None               | Medium          | 1          | 2                      | 3                      |            | 10                         | 0                      | 2                      | LowtoNone               |             |           |        |
| 2000 | 3           | 3       | 3       | 3       | 4        | 5       | 3       | 2       | 3       | 3       | 4       | 4       | 4       | 5       | 4       | 4      | 2       | 1        | 5       | 3       |         | 324               | C              | Rover              | EDL             | Medium     | 0                      | 5                      | 5          | DualString-Warmbackup      | 10                     | 0                      | 1                       | LowtoNone   | 1735.4    |        |
| 1993 | 2           | 3       | 3       | 4       | 3        | 3       | 4       | 3       | 2       | 4       | 4       | 2       | 5       | 5       | 3       | 5      | 1       | 5        | 3       | 6       | 3       | 113.19            | C              | Rover              | EDL             | Simple     | 3                      | 0                      | 3          | SingleString               | 10                     | 0                      | 1                       | LowtoNone   | 1080      |        |
| 2007 | 3           | 2       | 4       | 3       | 3        | 4       | 4       | 3       | 4       | 3       | 3       | 4       | 4       | 4       | 4       | 3      | 3       | 3        | 5       | 3       |         | 597               | C              | Rover              | EDL             | Complex    | 0                      | 10                     | 10         | DualString-Warmbackup      | 0                      | 1                      | 1                       | LowtoNone   | 1705      |        |
| 2000 | 3           | 3       | 4       | 4       | 3        | 4       | 4       | 3       | 3       | 4       | 4       | 3       | 4       | 5       | 4       | 5      | 2       | 4        | 3       | 3       | 6       | 3                 | 355.1          | C                  | Rover           | EDL        | Medium                 | 2                      | 5          | 7                          | DualString-Warmbackup  | 5                      | 1                       | 1           | LowtoNone | 1392.5 |
| 1998 | 4           | 3       | 3       | 4       | 4        | 4       | 3       | 2       | 3       | 3       | 2       | 3       | 3       | 3       | 5       | 3      | 4       | 2        | 3       | 6       | 3       | 50                | C              | DeepSpace          | SampleReturn    | Simple     | 2                      | 1                      | 3          | DualString-Coldbackup      | 5                      | 0                      | 0                       | Veryhigh    | 637       |        |
| 2002 | 3           | 2       | 3       | 3       | 3        | 4       | 5       | 3       | 3       | 3       | 3       | 5       | 4       | 4       | 3       | 4      | 3       | 3        | 2       | 6       | 3       | 224               | C/C++/Assembly | DeepSpace          | None            | Complex    | 0                      | 7                      | 7          | DualString-Coldbackup      | 1                      | 0                      | 0                       | Medium      | 691       |        |
| 2005 | 3           | 2       | 2       | 3       | 4        | 4       | 4       | 3       | 2       | 3       | 3       | ?       | ?       | ?       | ?       | ?      | ?       | ?        | 3       | 5       | 3       | 108               | C              | Observatory        | None            | Simple     | 3                      | 1                      | 4          | DualString-Coldbackup      | 1                      | 0                      | 0                       | Medium      | 446       |        |
| 1995 | 4           | 3       | 3       | 4       | 4        | 4       | 3       | 2       | 3       | 3       | 2       | 3       | 3       | 3       | 5       | 3      | 4       | 2        | 3       | 6       | 3       | 50                | C              | DeepSpace          | SampleReturn    | Medium     | 2                      | 0                      | 2          | DualString-Coldbackup      | 3                      | 0                      | 0                       | Veryhigh    | 546       |        |
| 1997 | 4           | 3       | 3       | 4       | 4        | 4       | 3       | 2       | 3       | 3       | 2       | 3       | 3       | 3       | 5       | 3      | 4       | 2        | 3       | 5       |         | 60.5              | C/Jovial       | DeepSpace          | None            | Medium     | 2                      | 1                      | 3          | DualString-Coldbackup      | 5                      | 0                      | 0                       | Veryhigh    | 336       |        |
| 2009 | 4           | 2       | 3       | 3       | 4        | 4       | 3       | 2       | 3       | 3       | 2       | 4       | 3       | 3       | 5       | 4      | 4       | 2        | 3       | 5       | 3       | 170               | C              | DeepSpace          | None            | Medium     | ?                      | 8                      | 8          | DualString-Coldbackup      | 0                      | 1                      | 0                       | Medium      | 552       |        |
| 1997 | 4           | 3       | 3       | 3       | 4        | 4       | 3       | 2       | 3       | 3       | 2       | 3       | 3       | 3       | 5       | 3      | 4       | 2        | 3       | 5       |         | 60.5              | C              | DeepSpace          | None            | Medium     | 2                      | 1                      | 3          | DualString-Coldbackup      | 3                      | 0                      | 0                       | Veryhigh    | 546       |        |
| 1995 | 3           | 3       | 3       | 3       | 3        | 4       | 3       | 2       | 3       | 3       | 3       | 3       | 3       | 3       | 4       | 3      | 4       | 2        | 4       | 3       |         | 184               | C              | DeepSpace          | None            | Medium     | 2                      | 0                      | 2          | SingleString               | 2                      | 0                      | 0                       | Medium      | 1042.8    |        |
| 1998 | 3           | 3       | 3       | 2       | 3        | 4       | 5       | 3       | 2       | 3       | 3       | 3       | 3       | 3       | 4       | 3      | 4       | 5        | 3       | 3       |         | 253               | C              | Observatory        | None            | Medium     | 3                      | 0                      | 3          | DualString-Coldbackup      | 0                      | 2                      | 0                       | Medium      | 2519      |        |
| 2005 | 3           | 2       | 2       | 3       | 4        | 4       | 3       | 2       | 3       | 3       | ?       | ?       | ?       | ?       | ?       | ?      | ?       | ?        | 3       | 5       | 3       | 135               | C              | Earth/LunarOrbiter | None            | Simple     | 1                      | 0                      | 1          | SingleString               | 1                      | 0                      | 0                       | High        | 492       |        |
| 2007 | 4           | 2       | 2       | 3       | 3        | 4       | 3       | 3       | 4       | 5       | 4       | 3       | 4       | 3       | 3       | 3      | 3       | 3        | 3       | 3       |         | 104.6             | C              | DeepSpace          | None            | Medium     | 6                      | 3                      | 9          | DualString-Coldbackup      | 1                      | 0                      | 0                       | High        | 320       |        |
| 2007 | 3           | 2       | 2       | 3       | 3        | 4       | 3       | 3       | 3       | 3       | 2       | 4       | 3       | 3       | 3       | 3      | 3       | 3        | 3       | 3       |         | 173.4             | C              | Earth/LunarOrbiter | None            | Simple     | 4                      | 0                      | 4          | SingleString               | 1                      | 0                      | 0                       | High        | 329       |        |
| 2010 | 4           | 2       | 4       | 3       | 4        | 4       | 3       | 2       | 3       | 3       | 3       | 4       | 2       | ?       | ?       | ?      | ?       | ?        | 5       | 3       | 3       | 155               | C              | Earth/LunarOrbiter | None            | Simple     | 2                      | 1                      | 3          | SingleString               | 1                      | 1                      | 1                       | Medium      | 789       |        |
| 2006 | 3           | 2       | 3       | 3       | 3        | 4       | 3       | 3       | 3       | 3       | 3       | 3       | 3       | 3       | 4       | 3      | 3       | 5        | 3       | 3       |         | 164.2             | C              | Earth/LunarOrbiter | None            | Medium     | 3                      | 0                      | 3          | SingleString               | 1                      | 0                      | 0                       | Medium      | 640.5     |        |

Deep Space

Earth/Moon



# Conclusions and Next steps

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ★ Initial results very promising:
  - ★ Reasonably accurate LOC estimators for early lifecycle data
  - ★ Effort estimators for early lifecycle data
- ★ Barry Boehm at the USC Center for Systems and Software Engineering is working with us and applying these methods on the COCOMO II data set
- ★ Next Steps under consideration
  - ★ Expand and improve SC flight software data set and add Instrument flight software
  - ★ Test with SEER-SEM
  - ★ 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



# *Back Up Slides*



# System Descriptors - 1

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

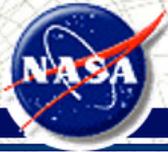
## System Descriptors

| Mission Type      | Values                      | Description   | Example   |
|-------------------|-----------------------------|---|---|
|                   | Earth/Lunar Orbiter         | Robotic spacecraft that orbit the earth or moon conducting science measurements. These spacecraft are very similar if not identical to the many commercial satellites used for communication as well as many military satellites. They often can have high heritage and even use production line buses from industry. | Aqua  |
|                   | Telecomm Sat                | Earth orbiters that support very high bandwidth and designed for very long life.  | TDRS  |
|                   | Observatory                 | Observatories are space based telescopes that support space based astronomy across a wide set of frequencies. They can be earth orbiters or earth trailing at the various lagrange points created by the gravoty fields of the earth, sun and moon.   | Hubble  |
|                   | Deep Space                  | Any robotic sapcecraft that goes beyond the moons orbit. So this category includes any misison whose destination is a planet, planetoids, any planetary satellite, comet, asteroid or the sun. These misison can be orbiters or flybys or a mixture of both.  | Deep Impact   |
|                   | Static Lander               | A robotic spacecraft that does its science in-situ or from the surface of a soplar system body. It does not move from its original location.  | Phoenix   |
|                   | Rover                       | A robotic spacecraft that does its science in-situ or from the surface of a solar system body and has the ability to move on the surface. To date all rovers have wheels but in the future they may crawl, walk or hop.   | Mars Exploration Rover (MER)  |
| Secondary Element | Values                      | Description   | Example   |
|                   | None                        | No secondary element  | Mars Reconsance Obiter (MRO)  |
|                   | Probe or Impactor           | A simple impactor with little or no guidance and navigation capabilityand once released it simply transmits data from its instruments A moderate-complexity impactor which may receive commands after separation, may have some internal guidance control, and several moderately complex instruments.                | Cassini-Huygens was a simple probe. Deep Impact had a medium complexity probe.    |
|                   | Entry, Descent, and Landing | EDL can be simple with a ballistic trajectory or complex with precision landing and hazard avoidance. All landers and Rovers will have an EDL element.  | Mars Pathfinder is an example of a simple EDL. MSL is an example of a complex EDL |
|                   | Sample return               | A simple sample return is a like a simple probe but returning to earth. A complex sample reurn would be a return from a planet surface and requires an ascent stage.  | Stardust is an example of a simple sample return                                  |



# System Descriptors -2

| Over All Complexity        | Values                     | Description   | Example                  |
|----------------------------|----------------------------|---|--------------------------|
|                            | Simple, Medium, or Complex | These are based on the mission type and secondary element so are derived from the descriptions above  | N/A                      |
| Number of Instruments      | Values                     | Description   | Example                  |
|                            | Simple                     | Any science instrument for which the FSW need only pass through commands and receive and store telemetry.   | Magneometer              |
|                            | Medium                     | Any science instrument for which the FSW must provide control logic that is relatively simple and requires no or only loose real time control. E.g., MER instruments.   | MER Instruments          |
|                            | Complex                    | Any science instrument for which the FSW must provide control logic that is complicated or requires tight real-time control.  | Telescope                |
| Flight Computer Redundancy | Values                     | Description   | Example                  |
|                            | Single String              | Spacecraft has no redundancy in the flight computer   | Most Earth Orbiters      |
|                            | Dual String - cold backup  | Spacecraft has redundant flight computers. Backup is normally off, is powered up and boots when prime string goes down  | Most Deep space missions |
|                            | Dual String - warm backup  | Backup computer is powered on and monitoring state of prime computer, but does not need to maintain continuous operation (e.g., a sequence may be restarted, attitude control restarts with last known state, etc.) | MSL                      |



# System Descriptors -3

| Number of Deployables | Values      | Description   | Example  |
|-----------------------|-------------|---|--|
|                       | Simple      | Simple deployable(s) which activate one time and remain in the deployed position for the duration of the mission.   | Magnetometer boom  |
|                       | Medium      | Moderately complex deployables which require some sequencing of deployment events, or may require deployment and retraction.  | Deployable Solar arrays  |
|                       | Complex     | Complex deployables with detailed deployment sequences, many deployments and retractions which may require additional control algorithms to compensate for changing system characteristics, or deployables which are critical to mission safety and/or success. | Parachute, bag inflation and retraction, rover standup, ramp extension, complex robotic arms). |
| Inheritance           | Values      | Description   | Example  |
|                       | Low to None | Software to be inherited has never flown in space. Significant new design and basically all new code.   | Mar Pathfinder or MSL EDL software   |
|                       | Medium      | Basic design has been used before but significant portion is new design and a code is newly written.  | MSL  |
|                       | High        | Software to be inherited has flown in space and performed satisfactorily. Inherited SW architecture but majority of code is newly developed.  | Many Planetary orbiters  |
|                       | Very High   | Software to be inherited was developed as a product line, has flown successfully in space at least once, has been successfully re-used in at least two missions, and has extensive documentation.   | Many earth orbiters  |



# Spectral Clustering

Cost Analysis Division

Jet Propulsion Laboratory

West Virginia University

- ✦ Select measure of distance
- ✦ Pick point A at random (near middle works better)
- ✦ Find furthest point from A (B)
- ✦ Find Furthest point for B (C)
- ✦ Draw line B-C
- ✦ Project all points onto the line and find the median. This is first eigenvector.
- ✦ Split data set by median point
- ✦ Repeat on each subset

