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
- 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

<table>
<thead>
<tr>
<th>COCOMO Inputs</th>
<th>Effort</th>
<th>LOC</th>
<th>Mission Descriptors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Dense</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Sparse</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
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.


---

MRE can distinguish between the models:

- Plotting the absolute values of the relative error, it is easily seen that Model 3 fits its data the best just as intuition would indicate.

$$MRE = \text{Magnitude of Relative Error, abs(Predicted - Actual)/Actual}$$
Data Mining Methods

- Data mining techniques provided us with the rigorous tool set we needed to explore the many dimensions 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


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

- **Size Distribution**
- **SLOC Range Estimate**
- **COCOMO Multiplier Range**
- **Spectral Clustering Effort Estimate**

**Clusters**

**Mission Descriptors**

**Model developed for this task**

### 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 overall MRE
  - **Median distance between two clusters is best**
    - Produces lower overall MRE
SLOC Estimation

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

![Graph showing median and mean MRES of LDEU Prediction]

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

![Graph comparing IMC, IMC_2PAIR, and IMC_2PAIR without outliers]

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