Jet Propulsion Laboratory
California Institute of Technology

ADVANCED STATISTICAL METHODS IN SPACECRAFT FLIGHT SOFTWARE COST ESTIMATION

Bayesian Regression and Nonlinear Principal Components Analysis to support System Engineering in the Early Project Lifecycle

Samuel R. Fleischer[1*]
Jairus M. Hihn[1]
James K. Johnson[2]

[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA

[2]National Aeronautics and Space Administration, Washington, DC

INCOSE 2022 International Symposium
June 25-30, 2022, Detroit, Michigan

*Corresponding author (Samuel.R.Fleischer@jpl.nasa.gov)

Agenda

- Who I am and what I do at NASA JPL
- Early formulation at JPL and the motivation behind building models accessible via the web
- Analogic and parametric models in early-project cost estimation
- The Analogy Software Cost Tool (ASCoT)
  - *Parametric models*
  - *Analogic models*
  - *Tool development and deployment*
- The Online NASA Space Estimation Tools (ONSET)

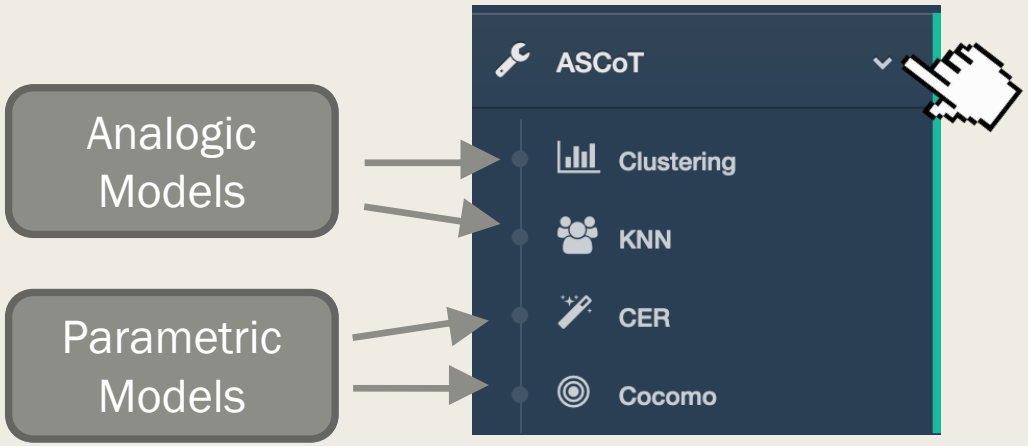
Early formulation at JPL, and the motivation behind building models accessible via the web

- Aspiring Principal Investigators with a glint in their eye come to JPL with ideas and want expert opinion on feasibility.
  - *These ideas have varying maturity.*
  - *They'll come back to JPL multiple times to get more refined cost estimates as the concept matures.*
- More NASA proposal calls = more concept design and trade studies at early stages. Models must be:
  - *Easily run by system engineers who are not domain experts*
  - *Transparent in data and algorithm*
  - *Easily accessible*
  - *Easily updated and distributed*

Analogic and Parametric models in early project cost estimation

- As the costing community in industry continues to refine parametric models, academia is focused on better understanding expert judgement and analogy.
- Early in the project lifecycle, inputs to parametric models are often loosely constrained, and thus parametric model output is of little use.
- It is instead useful to understand some measure of similarity to previous missions, and to form early cost estimates as analogies to these missions.
- As the concept matures, parametric models become more useful.
- Eventually, grassroots estimates outperform parametric models.

The Analogy Software Cost Tool (ASCoT) Overview



The data:

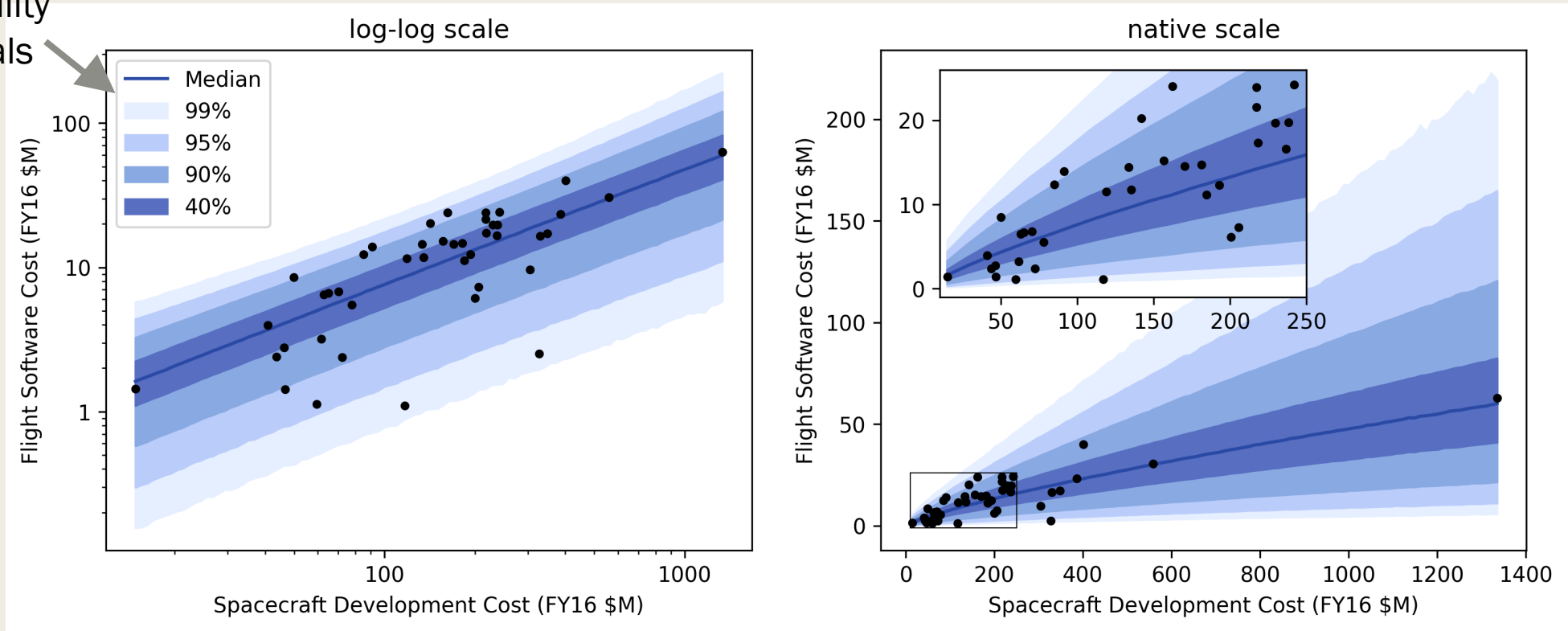
- N = 54 previously flown missions
- Sources
  - NASA CADRe
  - JPL SMART repo
  - project documentation
  - direct interviews
  - Independent, industry-wide dataset
- Variables
  - Destination
  - Redundancy
  - Software inheritance
  - Mission type
  - Mission size
  - Number of instruments
  - Number of deployables
  - Flight Software Cost ($)
  - Spacecraft Bus Cost ($)
  - Effort (WM)

COCOMO-II

- COCOMO (Constructive Cost Model) is a well-known parametric software cost model with 22 input parameters including
  - *Precedentedness (a measure of novelty of the mission)*
  - *Team Cohesion (a measure of consistency of stakeholder objectives and cultures)*
  - *Required Software Reliability (a measure of the effect of software failure)*
  - *Programmer Capability (A measure the abilities of project programmers)*
  - *and others*
- Model form: $\text{Effort} = (A \cdot M) \cdot S^B$ (Effort in work-months)
  - A: a measure of the baseline organizational and technological costs
  - M: a measure of the environmental factors
  - S: number of EqSLOC (logical equivalent source lines of code)
  - B: a measure of the economies or diseconomies of scale

The ASCoT Bayesian CER

credibility intervals



$$\log(\text{Software Cost}) = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$
$$\epsilon \sim \text{SkewNormal}(0, \sigma, \alpha)$$

Priors

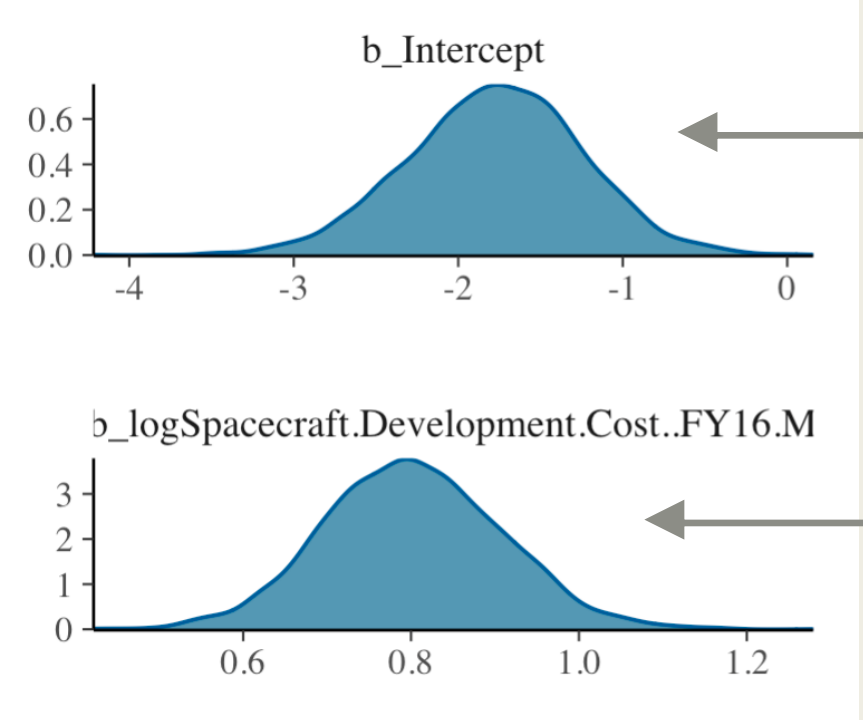
$\alpha \sim N(0,4)$

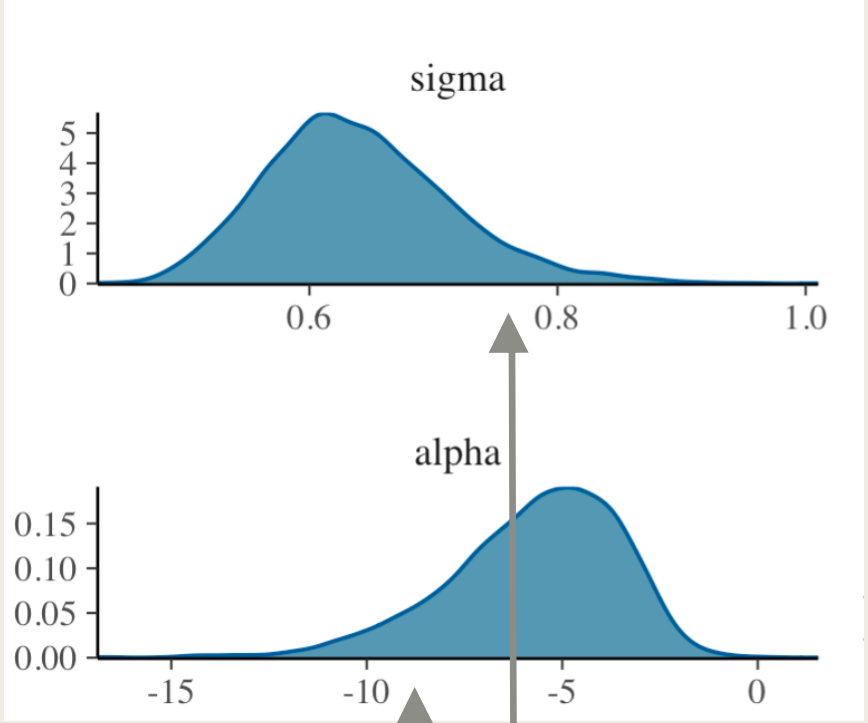
$\sigma \sim t(3,0,2.5)$

$\beta_0 \sim t(3,2.5,2.5)$

$\beta_1 \sim U(-\infty, \infty)$

Bayesian CER





$$\log(\text{Software Cost}) = \beta_0 + \beta_1 \log(\text{Spacecraft Cost}) + \epsilon$$
$$\epsilon \sim \text{SkewNormal}(0, \sigma, \alpha)$$

Priors

$\alpha \sim N(0,4)$

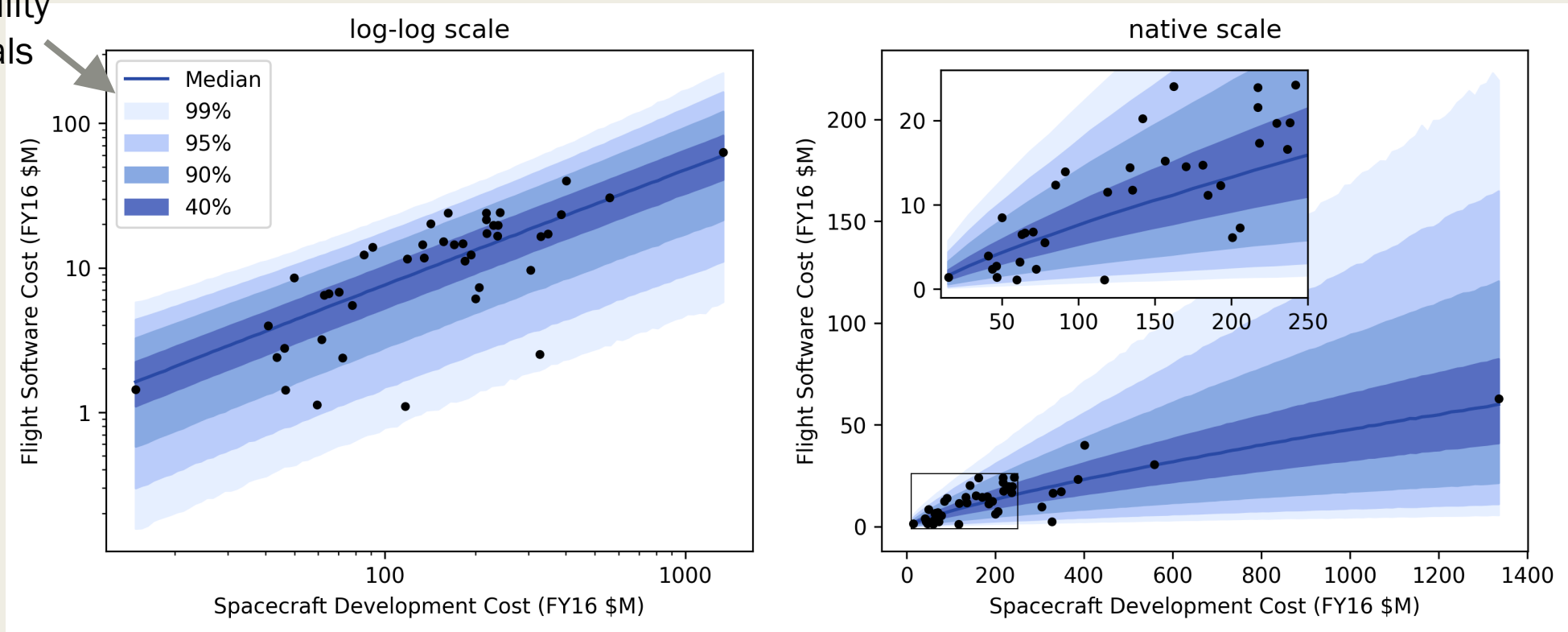
$\sigma \sim t(3,0,2.5)$

$\beta_0 \sim t(3,2.5,2.5)$

$\beta_1 \sim U(-\infty, \infty)$

The ASCoT Bayesian CER

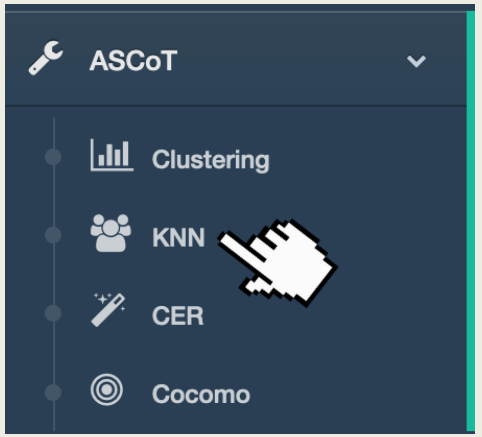
credibility intervals



- Skew normal error term performs better than normal error (log-skew-normal vs log-normal)
  - Captures low outliers without pulling median prediction down
  - Reasonable uncertainty bounds on the native scale due to negative skew
- Simple regression **performs better** than models including other software cost drivers such as number of instruments, destination, or redundancy **(TL;DR: avoids overfitting)**

ASCoT Analogic Models

- *k*NN model
  - *Finds the three missions most similar to your input*
  - *Estimate is a weighted average of these nearest neighbors*
- Cluster model
  - *Finds the mission cluster most suited to your input*
  - *Estimate is a weighted average of missions in your cluster*
- Both models utilize nonlinear principal components analysis (NLPCA)



NLPCA motivation

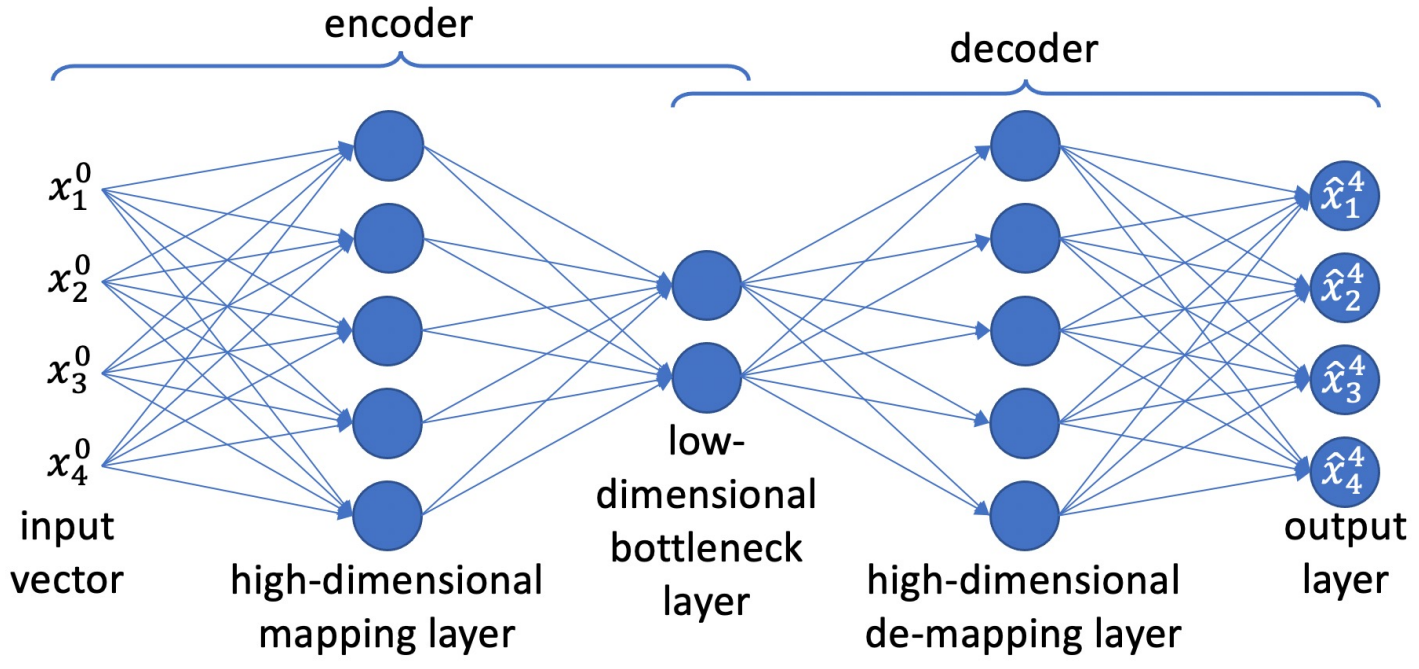
- How do we determine proximity of data when the data is numeric?
  - *Use a distance formula (Pythagorean or other)*
  - *Example*
    - Mission 1: (4 instruments, 5 deployables)
    - Mission 2: (2 instruments, 1 deployable)
    - Distance: $d = \sqrt{(4-2)^2 + (5-1)^2} = \sqrt{20}$
- How do we determine proximity of data when the data is NOT numeric?
  - *Example*
    - Mission 1: (Mars-bound, dual-string cold backup)
    - Mission 2: (Saturn-bound, dual-string warm backup)
    - Distance: $d = ???$

NLPCA motivation

- We have to find a way to numericize the data.
  - *Previous ASCoT versions chose "intuitive" transformations.*
  - *i.e. Single-string = 1, Dual-string (cold) = 2, Dual-string (warm) = 4.*
  - *This encapsulates the industry knowledge that the difference between a dual-string (warm) system and a dual-string (cold) system is greater than the difference between a dual-string (cold) system and a single-string system.*
- NLPCA lets the data speak for itself – optimal transformations are learned using machine learning… in particular auto-associative neural networks

NLPCA - ANNs

Auto-associative neural network



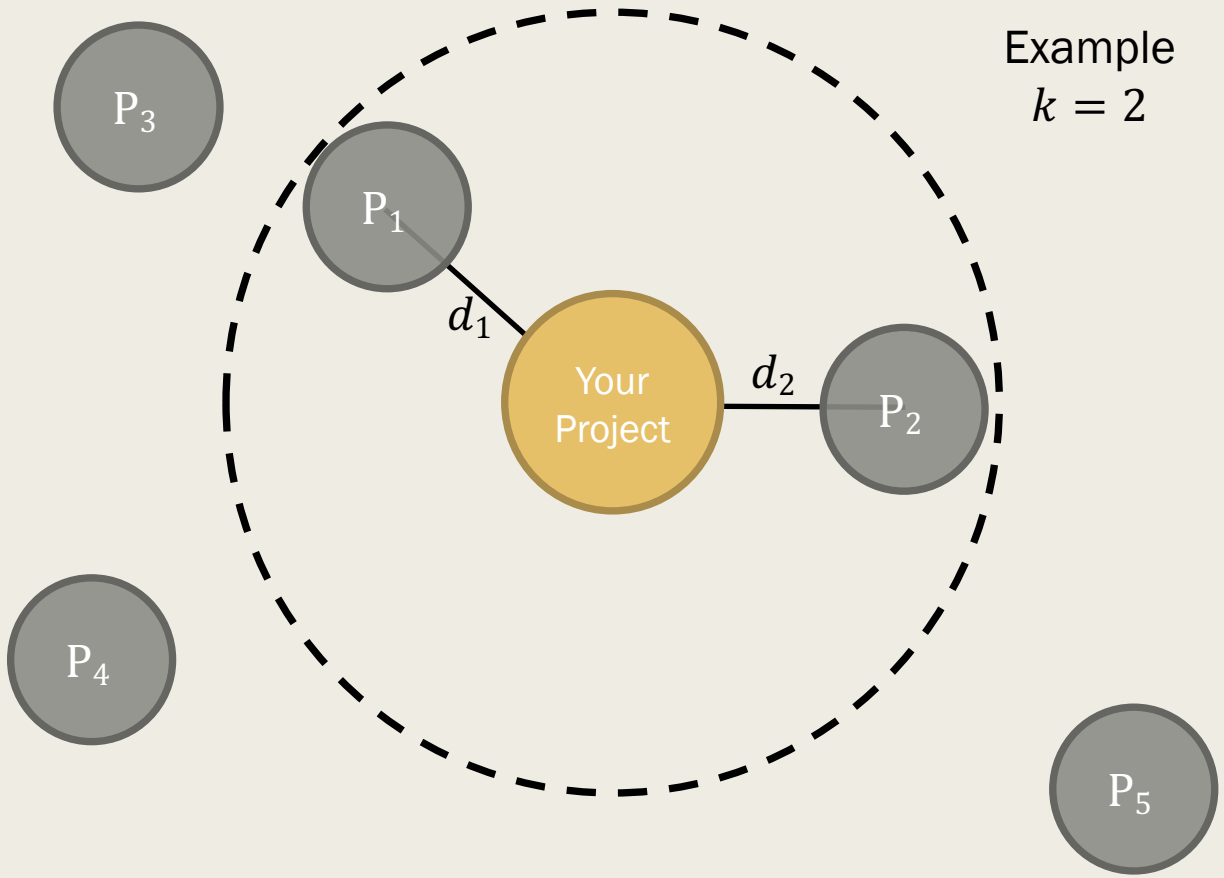
ANN parameters are optimized such that the difference between the output layer and the input layer is minimized.

The goal is for the low-dimensional bottleneck layer to adequately retain the information contained in the input layer.

The result is that a non-numeric input layer can be projected onto a numeric, low-dimensional space.

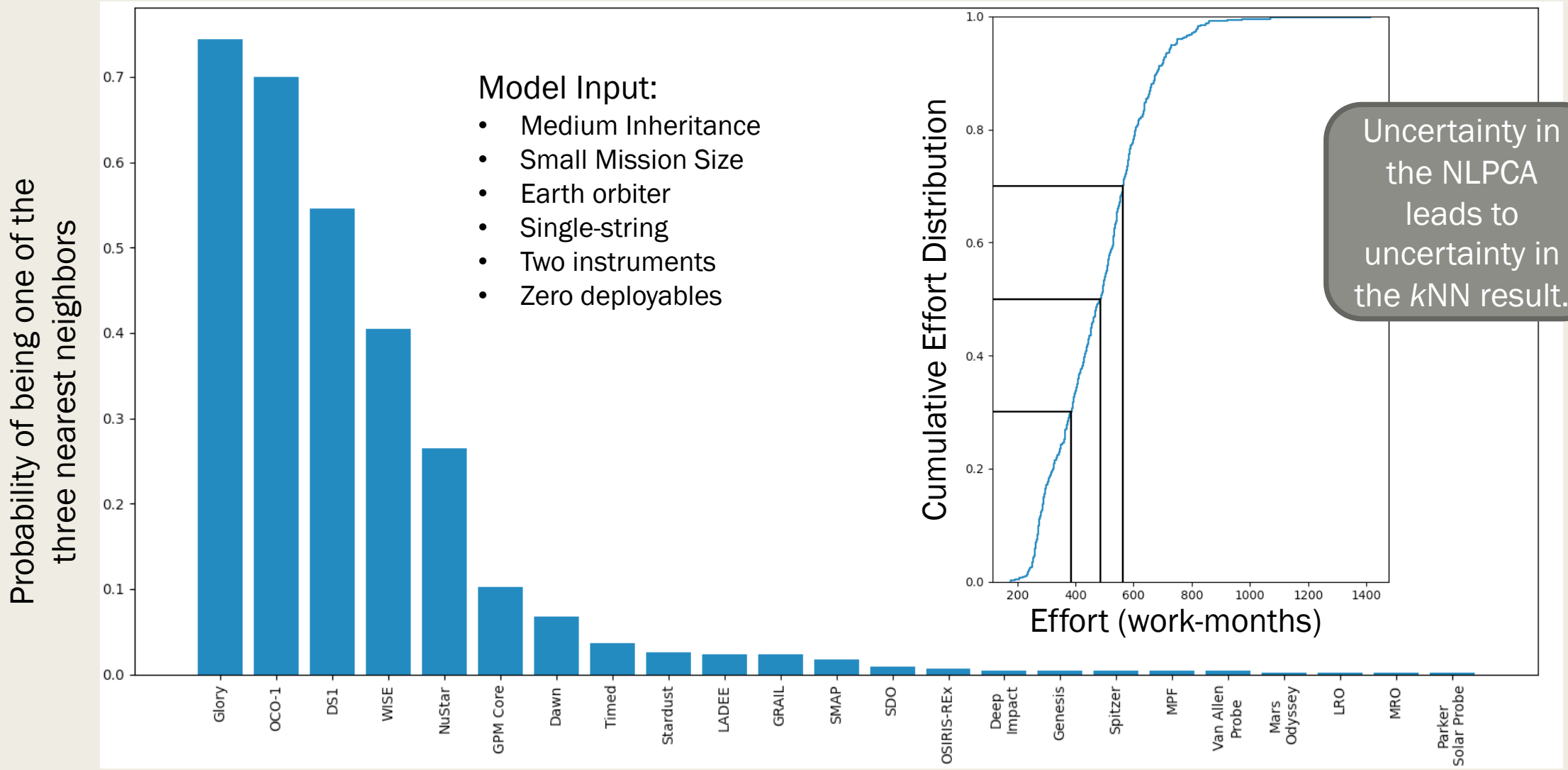
*k*NN Algorithm Overview

- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to any model input easily (in a well-defined manner)
- If we choose k=2, we only use the closest two missions to generate an estimate.



$$\text{Cost(Your Project)} = \frac{\frac{\text{Cost}(P_1)}{d_1} + \frac{\text{Cost}(P_2)}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}}$$

*k*NN Model Example Output

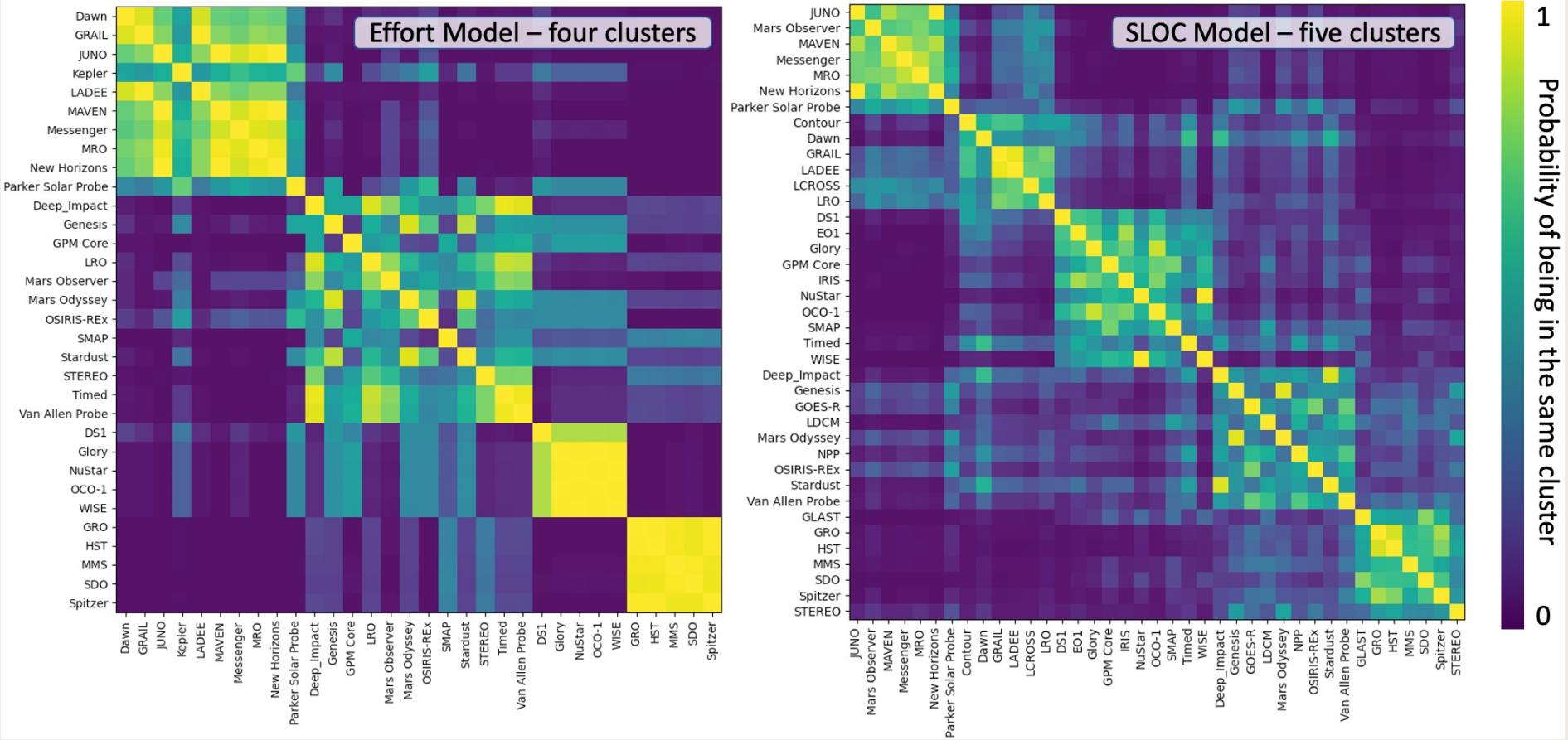


NLPCA-based Clusters

Effort Model Clusters						
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Complex, Earth-Inner Planetary	6. Smaller, Higher Inheritance	7. Large, Earth Observatories and Constellations
Cassini	MER	Insight	Dawn	Deep Impact	DS1	GRO
Galileo	MPF	Phoenix	GRAIL	Genesis	GLORY	HST
	MSL		JUNO	GPM Core	NuStar	MMS
			Kepler	LRO	OCO-1	SDO
			LADEE	Mars Observer	WISE	Spitzer
			MAVEN	Mars Odyssey		
			Messenger	OSIRIS-REx		
			MRO	SMAP		
			New Horizons	Stardust		
			Parker Solar Probe	STEREO		
				TIMED		
				Van Allen Probe		

SLOC Model Clusters							
1. Very Large, Old, Outer Planetary	2. Rovers	3. Landers	4. Large, Complex, Inner-Outer Planetary	5. Large, Moderately Complex, Dual String (Cold)	6. Smaller or Simple, Earth – Asteroid/ Comet	7. Small-Medium, Single-String Inner-Planetary or Dual String (Cold) Asteroid/Comet	8. Large, Earth Observatories and Constellations
Cassini	MER	Insight	JUNO	Deep Impact	DS1	Contour	GLAST
Galileo	MPF	Phoenix	Mars Observer	Genesis	EO1	Dawn	GRO
	MSL		MAVEN	GOES-R	GLORY	GRAIL	HST
			Messenger	LDCM	GPM Core	LADEE	MMS
			MRO	Mars Odyssey	IRIS	LCROSS	SDO
			New Horizons	NPP	NuStar	LRO	Spitzer
			Parker Solar Probe	OSIRIS-REx	OCO-1		STEREO
				Stardust	SMAP		
				Van Allen Probe	TIMED		
					WISE		

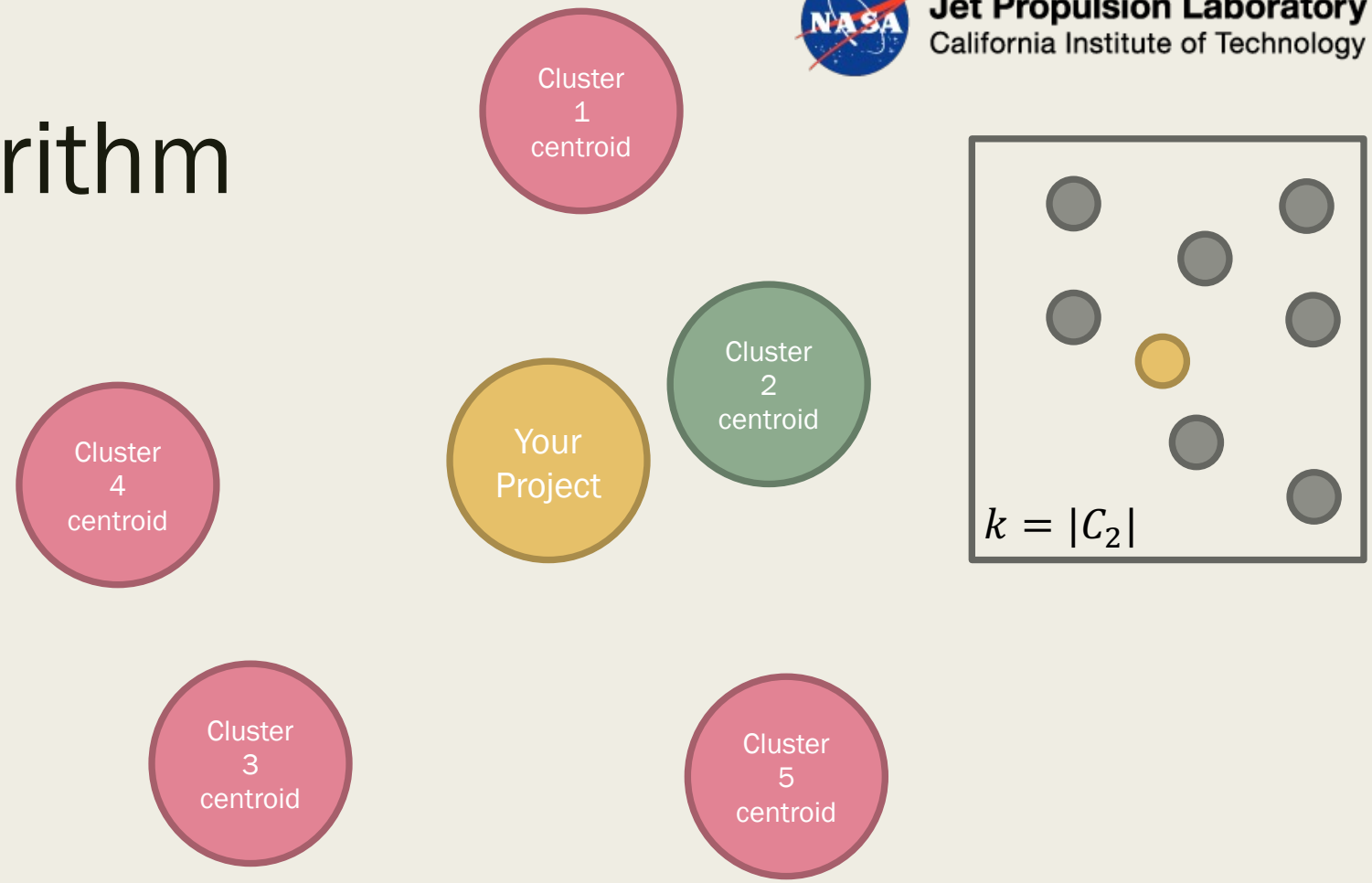
Clustering Algorithm Overview



Probabilistic Linkage Matrices
Calculated using the *k*-Means algorithm in NLPCA space
Cassini, Galileo, and Rovers and Landers are removed.

Clustering Algorithm Overview

- Once we have our missions in a low-dimensional numeric space, we can calculate the distance from each mission to the "center" of any cluster
- Once in a cluster with *k* missions, use the *k*NN weighted average formula for the estimate.



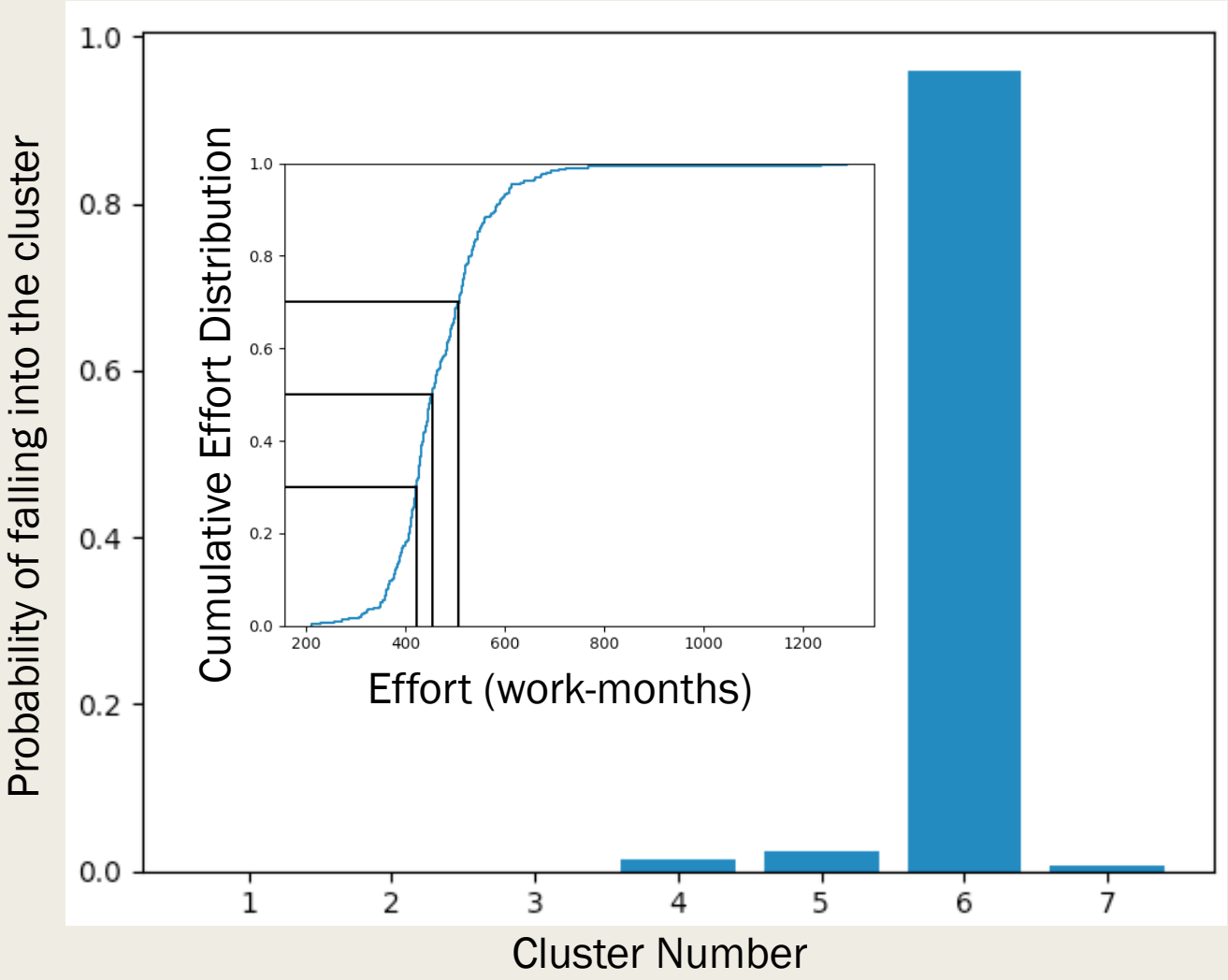
$$\text{Cost(Your Project)} = \frac{\sum_{P \in C_2} \frac{Cost(P)}{d(P, \text{Your Project})}}{\sum_{P \in C_2} \frac{1}{d(P, \text{Your Project})}}$$

Clustering Model Example Output

Model Input:

- Medium Inheritance
- Small Mission Size
- Earth orbiter
- Single-string
- Two instruments
- Zero deployables

Cluster 6 (Smaller, Higher Inheritance)
DS1
GLORY
NuStar
OCO-1
WISE



Uncertainty in the NLPCA leads to uncertainty in the cluster result.

Uncertainty in the Effort distribution is caused by uncertainty in the NLPCA as well as uncertainty in the cluster.

ASCoT as a web tool hosted on the Online NASA Space Estimation Tools (ONSET)

- ASCoT as a web tool is
  - *a set of four independent Dash (Python) applications, which each access…*
  - *the ASCoT master database*
- ONSET is a web framework
  - *Secure, Django framework, currently hosts two tools (built in Dash):*
    - ASCoT, and
    - COMPACT (the CubeSat Or Microsat Probabilistic and Analogies Cost Tool)
  - *Hosted independently behind the JPL firewall and by NASA HQ on ONCE (One NASA Cost Engineering)*
  - *Surprisingly simple (but with a learning curve) to build another tool from the ground up and include it in the ASCoT framework*

Stuff I didn't talk about today

- Previous version of ASCoT reported point estimates (instead of probability distributions)
  - *This is an example of an extended effort at JPL to improve Uncertainty Quantification (UQ)*
- Future of ASCoT includes parametric and analogic cost models of Instrument Flight Software (not just full flight software)
- If you're part of the NASA community and have a systems model you'd like to share as a web tool, let's talk!

Closing Out

- COCOMO, CER, kNN, and Clustering models all produce probabilistic output
- CER tool reports uncertainty in model parameters
  - *Full posterior distribution available for download as a CSV from the web tool*
- kNN and Clustering models utilize NLPCA and accounts for uncertainty in neural network fit
- If you don't have access to ONCE, you can request access by navigating to:
  - http://oncedata.com/ONCEUserAccessRequestForm.pdf
- Questions? Comments? Suggestions? Shoot me an email.
  - Samuel.R.Fleischer@jpl.nasa.gov