Wouldn’t it be nice to use all of your data points: An introduction to NICE

For Presentation at ICEAA – May 2023

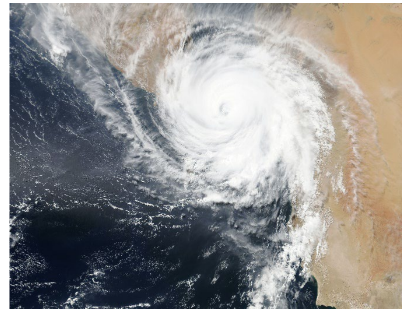
What is NICE?

N - Nonlinear

I - Iterative

C - Constrained

E - Estimator



What is NICE?

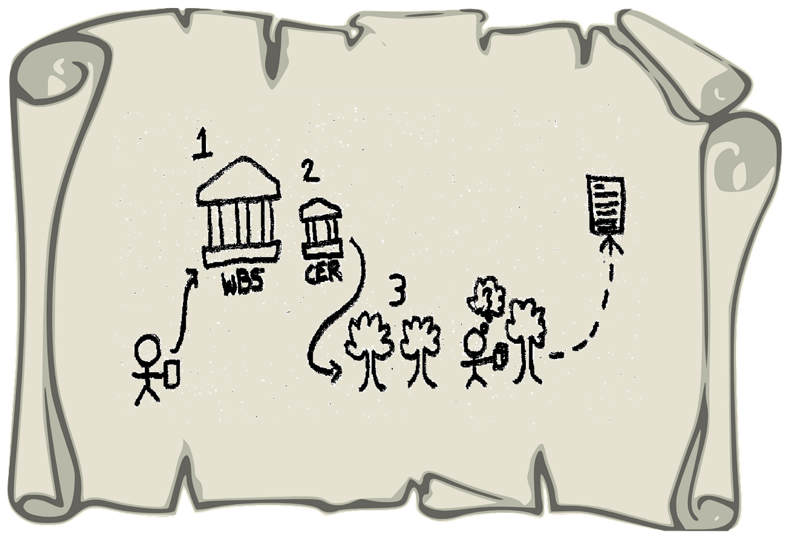
Where do we begin?

The Agency is building a new Special Vehicle and your job is to provide the cost estimate.

1. You visit the WBS library and build out a WBS to describe the new vehicle along with important technical variables that define what makes the new vehicle special.
2. You then visit the CER annex of the WBS library to build equations to relate the technical variables to their component of the vehicle cost.
    1. Vehicle$ = a * Payload$ + b * Power
    2. Test$ = c * PMP$ + d * Endurance
3. The next stop in your adventure takes you to the Forest of Data Collection where you must search for analogous vehicles that can be used as part of your CER estimate that will be used to drive the overall vehicle cost estimate.

Example WBS

WBS
Development$
PMP$
Vehicle$
Payload$
Test$
Power
Endurance



What is the Problem?

While you are collecting data you are able to find 7 other vehicles that can be used in your estimate.

There is a problem with the data though, for the vehicles which match the special ability of the new vehicle important data values are missing.

Missing data means that some extra steps must be taken to fit the data into the estimate.

WBS	Vehicle1	Vehicle2	Vehicle3	Vehicle4	Vehicle5	Vehicle6	Vehicle7
Development$	16,281	1,176	4,614	711	520	4,768	11,600
PMP$	15,643	1,040		661	460		5,800
Vehicle$	11,543	820		439	356		3,600
Payload$	4,100	220	181	222	104	844	2,200
Test$	638	136		50	60		5,800
Power	13,001	1,200	940	1,501	900	5,200	15,000
Endurance	121	36	72	25	96	48	240

The problem is missing data

Presented at the ICEAA 2023 Professional Development & Training Workshop - www.iceaaonline.com/sat2023

Three roads diverged in a yellow wood

How do you handle the missing data?

1. Cut out the vehicles with missing data values. This will keep the CERs defined at the lowest level elements and continue to estimate from the bottom up in the WBS structure, but the CER may be biased because vehicles with special capabilities have been removed from the dataset.

2. Change the CER definitions to use variables which have all of their data, something like:
   Development$ = a*Payload$ + b*Power + c*Endurance + d
   This then requires a top down allocation scheme to spread overall costs to lower level WBS elements.

3. Use data imputation to fill in the missing values. The CER definitions remain the same, key vehicles are still included in the dataset, and the overall estimate is still computed from the bottom up.

Aside: Real World Situations

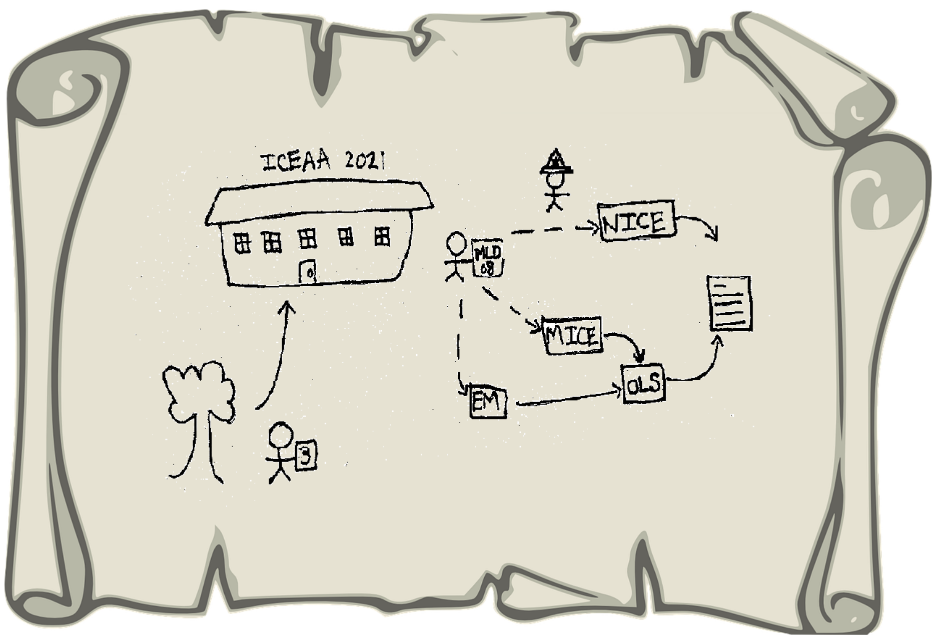
- Real world data can be missing data inconsistently
  1. Missing total system costs
  2. Only summary data available
  3. Missing technical descriptor

WBS	Vehicle1	Vehicle2	Vehicle3	Vehicle4	Vehicle5	Vehicle6	Vehicle7
Development$	16,281	1,176	4,614	711	520		11,600
PMP$	15,643	1,040		661	460		5,800
Vehicle$	11,543	820		439	356		3,600
Payload$	4,100	220	181	222	104	844	2,200
Test$	638	136		50	60		5,800
Power	13,001	1,200	940	1,501	900	5,200	15,000
Endurance	121	36	72	25	96	48	240

WBS	Vehicle1	Vehicle2	Vehicle3	Vehicle4	Vehicle5	Vehicle6	Vehicle7
Development$	16,281	1,176	4,614	711	520		11,600
PMP$	15,643	1,040		661	460		5,800
Vehicle$	11,543			439	356		3,600
Payload$	4,100		181	222	104	844	2,200
Test$	638			50	60		5,800
Power	13,001	1,200	940	1,501		5,200	15,000
Endurance	121	36	72	25	96	48	240

How do we finish the estimate?

- You have decided that you cannot remove any vehicles from the dataset and will utilize data imputation to perform your CER estimates.
- Luckily for you, ICEAA is hosting a presentation on "Dealing with Missing Data - The Art and Science of Data Imputation" (Roye, Hilton, and Smart) and from there you learn about Expectation Maximization (EM) and Multiple Imputation by Chained Equations (MICE).
- EM and MICE will fill in the missing data and then that full dataset can be used to run a CER analysis.



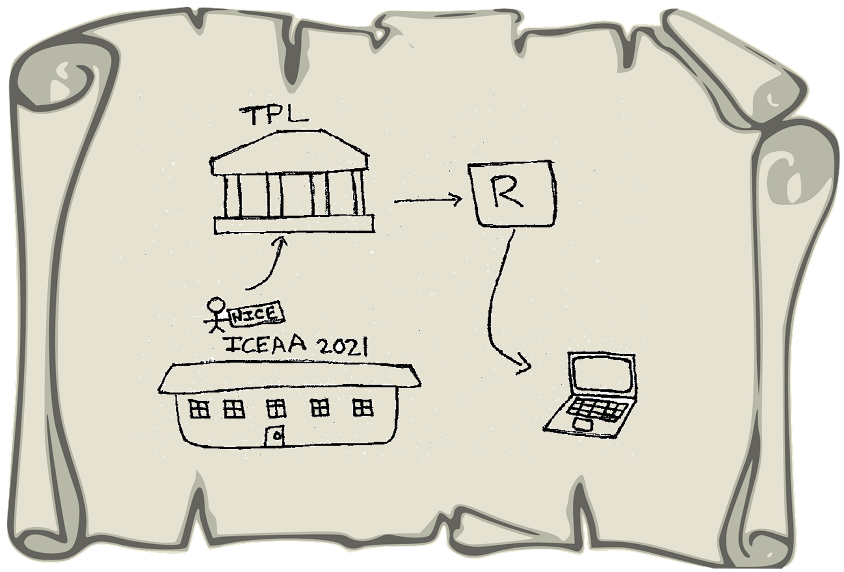
What Is NICE?

NICE is a tool to fit CERs when data is missing.

- Non-linear – NICE can fit CERs of any form and does not require them to be linear or log-linear
- Iterative – NICE uses an iterative method to search for the best solution
- Constrained – NICE uses WBS additivity constraints to fill in data and fit CERs
- Estimator – NICE generates numbers to use in your estimate

Origins of NICE

- I too attended ICEAA 2021 and went to the presentation on data imputation.
- After the presentation some colleagues mentioned that Tecolote also had a data imputation method which was used in the past.
- I started my own quest to discover how this method worked and whether it could be brought back to life for people to use again. This led me to the Tecolote Publication library, which was filled with many documents describing NICE.
- I took that information and now NICE has been reimplemented in R and is available to Tecolote analysts.



Using NICE: Building a NICE Model

NICE requires two sets of inputs

1. WBS and Technical Inputs (Model Inputs)
   - Initial Data values for WBS and Technical elements
   - Constraint matrices which define the WBS hierarchy
2. Equation Inputs
   - Initial Coefficient values used in the equations
   - Equation definitions

Model Inputs – Initial Values

- Initial values fill in missing data as a starting point for NICE
  - NICE will accept any kind of initial values, any invalid values will be corrected before starting the search for results.
  - Different search paths may lead to different results, take time to consider initial values which might favor specific outcomes.
  - These could favor Vehicle$ over Test$, or vice versa. They could also start without favoring either element.

LABEL	WBS	Vehicle 3 D_3	Vehicle 3 D_3	Vehicle 3 D_3	Vehicle 3 D_3
x_1	Development$	4,614	4,614	4,614	4,614
x_2	PMP$	0	4,613	182	2,397
x_3	Vehicle$	0	4,432	1	2,216
x_4	Payload$	181	181	181	181
x_5	Test$	0	1	4,432	2,217
x_6	Power	940	940	940	940
x_7	Endurance	72	72	72	72

- Choice of initial values can possibly change the results

Model Inputs – Constraint Matrices

- Each data point D_i has a constraint matrix C_i along with a vector B_i where $C_i * X = B_i$
- Two types of constraints
  1. Value override constraints
  2. Additivity constraints

Constraint Type	X_1	X_2	X_3	X_4	X_5	X_6	X_7	B
Value	1	0	0	0	0	0	0	4614
Additive	-1	1	0	0	1	0	0	0
Additive	0	-1	1	1	0	0	0	0
Value	0	0	0	1	0	0	0	181
Value	0	0	0	0	0	1	0	940
Value	0	0	0	0	0	0	1	72

1. Row 1: x1 = 4614
2. Row 2: -x1 + x2 + x5 = 0

Which we can substitute the WBS names to write as:

– Development$ + PMP$ + Test$ = 0

Which can be rewritten in a more familiar form as:

Development$ = PMP$ + Test$

Equation Inputs

- Initial coefficient values may come from a CER analysis with some data points removed, or previous studies on a similar topic. They are formatted as a single vector:

Coefficients
0.9
0.3
0.2
4

- Equations in NICE can relate the different model elements to each other and are defined in terms of the objective function
  - Equations:
    1. Vehicle$ = coef1 * Payload$ + coef2 * Power
    2. Test$ = coef3 * PMP$ + coef4 * Endurance
  - Additive error term:

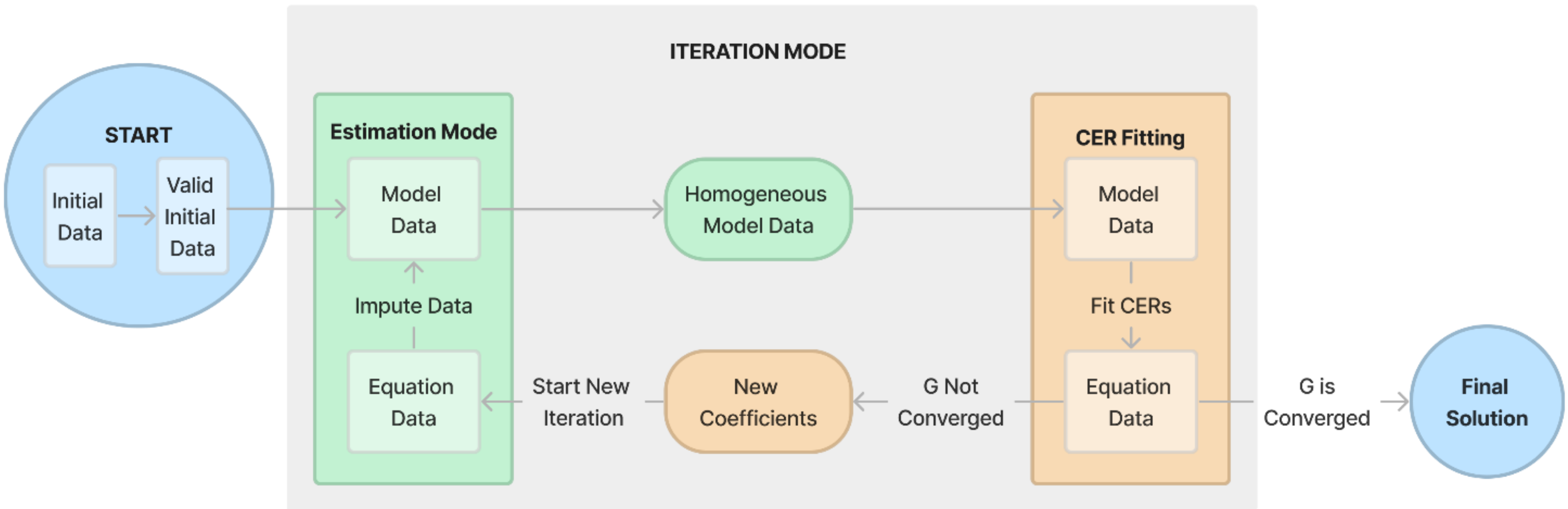
Equations
(x[3] - (x[4] * coef[1] + x[6] * coef[2]))^2
(x[5] - (x[2] * coef[3] + x[7] * coef[4]))^2

  - Multiplicative error term:

Equations
(log(x[3]) - log(x[4] * coef[1] + x[6] * coef[2]))^2
(log(x[5]) - log(x[2] * coef[3] + x[7] * coef[4]))^2

What does NICE do?

- Initial data is turned into valid initial data by NICE
- Estimation Mode – coefficients are fixed and new data values are imputed for missing data values in the model
- CER Fitting – data values are fixed and new coefficient values are calculated for the equations
- Iterate until converged, for convergence thresholds you define



What do the results look like?

NICE Mode:

○ Estimation ◉ Iterative

Best Fit Data:

WBS	D1	D2	D3	D4	D5	D6	D7
Development$	16281.00	1176.00	4614.00	711.00	520.00	4768.00	11600.00
----PMP$	15643.00	1040.00	1757.08	661.00	460.00	3199.12	5800.00
--------Vehicle$	11543.00	820.00	1576.08	439.00	356.00	2355.12	3600.00
--------Payload$	4100.00	220.00	181.00	222.00	104.00	844.00	2200.00
----Test$	638.00	136.00	2856.92	50.00	60.00	1568.88	5800.00
Power	13001.00	1200.00	940.00	1501.00	900.00	5200.00	15000.00
Endurance	121.00	36.00	72.00	25.00	96.00	48.00	240.00

Best Fit Coefficients:

Coefficients
3.74
-0.29
-0.12
24.14

NICE Opportunities

- Preserve Data – Use all of the data points and estimate a desired level.
- Testing Non-Linear Hypotheses – Each CER can be linear or non-linear.
- NICE is a framework – no specifics are required for non-linear search methods (Newton vs Simplex) or other elements of the process. There are many opportunities to customize how it works.
- Utilizing New Technologies – use of R and other mathematical programming languages lowers the barrier for entry.

NICE Limitations

- Using Imputed Data in Each Iteration – chaotic search path through the iterations
- Non-Linear Minimization Pitfalls – input sensitivity, the butterfly effect
- System Estimation – NICE estimates the entire system of equations and may make tradeoffs to the best fit compared to fitting a single equation
- Unconstrained- Minimization Pitfalls – potential nonsense values like negative costs

Topics For Further Study

- Multivariate CAIV
  - Problem definition for CAIV fits into NICE Estimation mode
  - Collaboration with real world data helps this effort
- NICE Advancements
  - Automatically cover initial input spaces to try and find a global best fit instead of just a local best fit
  - Automatically suggest ways to constrain undesirable outputs
  - Collaboration helps guide improvements to NICE and recommendations on when it is best used
- Suggestions?

Conclusion

- Data imputation allows for estimates at detailed levels and incorporating all of the data points
- NICE goes beyond just imputing data to incorporate additivity constraints as well
- Lessons from the past can still be useful for tackling problems today
- Collaboration is the key to moving forward, take the road less traveled by and see if that makes a difference for you

Consider working with us to apply NICE to your data

Contact

- Steven Read, Tecolote Research, Inc.
- Email: sread@tecolote.com