RT-206: Data science approaches to prevent failures in systems engineering

Sponsor: DASD(SE)

By
Prof. Karen Marais (AAE) and Prof. Bruno Ribeiro (CS)

11th Annual SERC Sponsor Research Review
November 19, 2019
FHI 360 CONFERENCE CENTER
1825 Connecticut Avenue NW, 8th Floor
Washington, DC 20009

www.sercuarc.org
Project failures occur despite systems engineering best practice

Data obtained from Project Management Institute (PMI): Pulse of the Profession 2018
Humans are an integral part of project failures

• People usually do not willfully cause failures but make mistakes that lead to failures
  – E.g. Get distracted while conducting equipment testing, communicate poorly during design phase

• Activities, behaviors, and personality can lead to poor practices and individual performance
  – E.g. Poor team coordination decreases productivity\(^1\)

• **Our core ideas:**
  – Risk assessment based on these human actions that lead to failures
  – Predict failures and get actionable insight as to how to avoid them
  – Have an adaptable process that is tailored to the organization

In 1904 Spearman published his “General Intelligence” Objectivity Determined and Measured

- Observed human ability as a manifestation of latent general intelligence factors

- Spearman’s paper was pinnacle of 20th century statistical tools: factor analysis

Influential because latent factors can explain why seemingly unrelated observations are correlated in practice

- Spoiler alert: They share hidden common “root causes”
Presentation Roadmap (Today)

• **Year 1** in 25 minutes:

*Capturing Latent Failure “Root Cause” Factors from Crowd Signals*

• **Q1:** How to get failure signals?
  A: Ask the crowd

  1. *Don’t ask if someone’s team is properly assessing risk,* ask
     • “Did you spend time discussing trivial matters with your team last week?”
     • “Did you disagree with an idea because you did not understand all the potential implications?”
     • “How often did you notice a “silent room”?”
     • ...
  2. Ask the all team members (crowd) rather than trusting a single individual
     • Rarely one person has a complete picture of a project

• **Q2:** How to extract latent failure “root cause” factors from crowd signals?
  A: *Latent factors + state-of-the-art machine learning tools*

  1. Spearman’s latent factor cannot be applied to new teams
  2. More predictive factors with less data: From 20th century Spearman to state-of-the-art machine learning
Year 2 Ideas:

1. *Causation from data alone*
   - From latent “root cause” factors → corrective action
   - Knowing latent “root causes” ≠ knowing how to intervene
   - Correlation is not causation
     • But causation may be inferred from correlation through novel machine learning tools

2. *Industrial Partnerships*: Student projects → industry projects
   - Deployment on “real” projects
     • Industrial partnerships
       - Year 1 student projects → Year 2 “real” industrial projects
     • Can we ask fewer questions without loss of power?
Why Crowd Signals?
A Crowd-Based Risk Assessment Tool for Organizations

**Easy to collect**
Collect data in a way that is not cumbersome, via an online crowd-signal app

**Adaptable**
Use data science and machine learning to make the approach adaptable to the organization

**“Hard-to-game”**
Questions that do not have obvious answers, but capture human behavior

**Frequent & expanded data collection**
We ask a large number of employees about the projects frequently, collecting risk information continuously

**Based on the common reasons of failure**
Previous research has identified 21 “real reasons” that projects fail (e.g., poor documentation)

**Human behavior at the core**
Include a variety of signals from factors that impact project and team performance
Capture “the pulse of the organization” frequently, continuously, and as efficiently as possible

- Crowd signals
  - Data collected directly from members of a project team
  - Questions that people answer about their behaviors and activities related to their projects
  - Start with a large set of questions, then narrow down
Student teams as testbed for industry teams

- Smaller-scale engineering projects allow students to design, manufacture, test, or operate equipment

- Students work in teams and make weekly progress with meetings and interaction with the instructor

- We have ready-access to such teams

- The teams are doing “real engineering”

- The students come from diverse backgrounds, reflecting what industry looks like
Literature as a guide to develop the crowd signals

- Factors that affect individual, team, and project performance
- Cognitive biases and safety archetypes
- Indirect metrics we suspect relate to failure
  - E.g. Personal habits, number of ordered parts, number of unscheduled team meetings
- 49 questions in total
- The students respond to these 49 questions every week
When asking people questions, there is a problem.

- **Bandwagon effect:** Tendency to do or believe what others do or believe.

- **Focusing effect:** Tendency to place too much importance on one aspect.

Do you suffer from bandwagon effect?

Do you suffer from focusing effect?

No, of course not!
• When possible, we phrase questions as hard to game and in context of a student project

Bandwagon effect

Focusing effect
Fixing symptoms rather than root causes:

Using symptomatic solutions that become less effective over time resulting in problem resurfacing
Instructors provide the failure events we need to train the predictive models

- Three qualitative metrics of project success
  - One question for each
  - Dependent variables

Team 3 is behind on schedule this week
We asked the instructors three questions, one for each project metric.

<table>
<thead>
<tr>
<th>Project</th>
<th>Spending Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>On budget</td>
</tr>
<tr>
<td>Project 2</td>
<td>On budget</td>
</tr>
<tr>
<td>Project 3</td>
<td>On budget</td>
</tr>
<tr>
<td>Project 4</td>
<td>On budget</td>
</tr>
<tr>
<td>Project 5</td>
<td>On budget</td>
</tr>
</tbody>
</table>

What is currently true about the project spending, compared to what you initially planned?
How to extract latent failure “root cause” factors from crowd signals?
Challenge: Spearman’s latent factors cannot be applied to new teams
Q1: Are the Crowd questions useful to predict failures?
Simple logistic regression models to predict failures

- Problem: To predict whether a student team will have a failure in the next week, based on their responses this week
  - Occurrence of failure is binary, so we use classification
  - Mixed effects model to account for correlation between responses from the same student
  - 3 models in total: budget, schedule, technical requirements failure
  - Logistic regression includes lots of assumptions, but hints to which questions correlate with which failure

\[
\log P[\hat{Y}_{i,t+1} = \text{Fail}|X_{it}] \propto a + bX_{it}^T + u_i + \epsilon_{it}
\]
Current Logistic Regression Model

Prediction of project outcome:

\[ P[\hat{Y}_{budget,t+1} = \text{Fail} | X] \]

\[ \sigma(\sum) \text{ sigmoid function} \]

Alice's variable
\[ u_{Alice} \]

How often did you notice a "silent room"? (time t)

\[ X \quad 10 - \text{"always"} \]

Did you spend time discussing trivial matters with your team last week?

yes

...
How often did you think your team made meaningful progress last week? (Likert-scale)

Record the change in number of project outputs from your team last week (Likert-scale)

Did you spend time discussing trivial matters with your team last week?

Record how many times you met with your team outside regular time

Based on p-values < 0.05

Requirements Failure

Was there a requirements failure last week? (from instructor)

Did you learn anything new this week?

What is your current estimate about your project’s technical performance?

How often did you notice a “silent room”? (Likert-scale)
\textit{k-fold cross-validation} to gauge the ability of the models to generalize

- \textit{K=10 folds}
  - Find accuracy in binary predictions (failure/no failure) from unknown data

\begin{itemize}
  \item\textbf{Testing fold}
  \item\textbf{Training fold}
\end{itemize}

\begin{itemize}
  \item Iteration 1
  \item Iteration 2
  \item Iteration 3
  \item \ldots
  \item Iteration 10
\end{itemize}

\begin{itemize}
  \item \textbf{Fold Accuracies}
  \item $e_1$
  \item $e_2$
  \item $e_3$
  \item \ldots
  \item $e_{10}$
\end{itemize}
Budget and schedule model are on average 54% accurate, requirements model is 60% accurate.
Q2: Can we find latent factors for new teams / new students?
Small Datasets: Information is precious, simple models are not data-efficient

- Improving accuracy & obtaining latent factors for new teams requires going from simple to complex:
  
  - Simple models (e.g., logistic regression, latent factor models)
    - Data-wasteful
    - Latent factors cannot be used in new teams / new students

  - Complex models (e.g., neural network factor models, boosted decision trees)
    - Data-efficient
    - Latent factors on new teams / new students
Simple models

e.g., logistic regression, (Spearman’s) latent factor models

Cons (simple):

- *Data-wasteful*: Cannot capture all information from data
- Either no latent factors or cannot apply method to new teams

Pros (simple):

- Will not overfit the training data
- Parameters can be “interpreted”
  - Not quite if model is wrong (misspecified)
Complex models

- **Pros (complex):**
  - *Data-efficient:* Will extract most information from data
  - Can obtain latent factors directly applied to new students, new classes

- **Cons (complex):**
  - *Will overfit the data*
    - Less of an issue with modern optimization methods
  - Interpretation can be more challenging

\textit{e.g., neural factor models}
Advanced Model

\[ P[\hat{Y}_{\text{budget},t+1} \mid \{X_{1,t}, \ldots, X_{4,t}\}] \]

Neural Network Factor Model

Student-specific effects

Group effects

Janossy pooling

Janossy pooling

Learning functions on sets inputs

Learning functions on set inputs

How often did you notice a "silent room"?

Did you spend time discussing trivial matters with your team last week?

<table>
<thead>
<tr>
<th></th>
<th>10 - &quot;always&quot;</th>
<th>10 - &quot;always&quot;</th>
<th>1 - &quot;never&quot;</th>
<th>1 - &quot;never&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 - &quot;always&quot;</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>10 - &quot;always&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - &quot;never&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - &quot;never&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Advanced Model is More Accurate than Logistic Regression
• Single task = just predict either budget, requirements, or schedule
• Multi-task = jointly predict budget, requirements, and schedule

*Advanced model generalizes better to test data*
(results from 5-fold cross-validation)
• Multi-task learning better than predicting single task
• Obtains latent “root cause” factors for new students

<table>
<thead>
<tr>
<th>Project Failure</th>
<th>Logistic Regression (Single Task)</th>
<th>Our Model (Single Task)</th>
<th>Our Model (Multi-task)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budget</td>
<td>0.642 ± 0.080</td>
<td>0.689 ± 0.09</td>
<td>0.729 ± 0.068</td>
</tr>
<tr>
<td>Schedule</td>
<td>0.523 ± 0.072</td>
<td>0.586 ± 0.068</td>
<td>0.580 ± 0.041</td>
</tr>
<tr>
<td>Technical Requirements</td>
<td>0.580 ± 0.062</td>
<td>0.643 ± 0.035</td>
<td>0.688 ± 0.088</td>
</tr>
</tbody>
</table>

± standard deviation
Year 1 resulted in 5 publications overall from the research team and one best conference paper award


ASEE Annual Conference and Exposition 2019

In this paper we compared the frequency of certain types of systems engineering failure causes occurring in industry vs. in student design projects.

Georgios Georgalis is part of the research team under Dr. Karen Marais
Year 2 Ideas
We are currently working on the feedback process

- Feedback has two parts
  - A prediction of budget, schedule, and requirements failure
  - Recommendations to prevent them (from a repository)
- But
  - Logistic regression/neural latent factor model captures correlation only
    - Recommendations are causal
  - Under multiple causes, causal model could be learned from our data
    - “The blessing of multiple causes” [Wang & Blei, JASA 2019]
Based on models we built with data from previous teams that received no feedback and the responses from your team members from last week: We predict that you have \( \hat{P}(Y_{1,t+1} = 1) \) chance of having a budget failure. To improve your team’s chances of success in terms of the budget, we suggest \( R_1 \), which will decrease the chance to \( \hat{P}(Y_{1,t+1} = 1) \).
Reducing the prediction model input size

• Simple **logistic regression** needs 49 questions as the input to predict the failures with reasonable accuracy

Would like to reduce the number of questions we ask:
  − Reduce the burden on the respondents
  − Make it more “realistic” as an application
  − We only care about the questions that matter (correlate with failure)
  − It will require advanced models
  − It will require causal models

• To accomplish this we are currently doing the following analyses:
  − Feature selection: get feedback from model on which questions are really relevant
  − Study advanced models that can still work well with fewer question
  − Study feature selection in causal models (new machine learning tools)
For Year 2 we propose to partner with an organization to expand our inputs and test in “real” projects.

- We will deliver value to our partners by leveraging the predictive capability of our prototype. Our process gives us the opportunity to provide feedback to decision makers, alerting them of upcoming failures, and suggesting corrective actions.