An Alternative Flight Software Trigger Paradigm

Kelly Smith, Robert Gay, Susan Stachowiak

NASA Johnson Space Center

AIAA GN&C, August 19-22, 2013, Boston, MA
• Orion is scheduled to fly an orbital test flight, Exploration Flight Test-1 (EFT-1), in late 2014

• Mission Profile:
  - Launch from Kennedy Space Center aboard Delta-IV Heavy into LEO parking orbit
  - After 1 orbit, Delta IV-H upper stage injects the vehicle into an elliptical orbit which intersects the Earth
  - Orion flies high-speed guided entry, splashes down in Pacific Ocean west of Mexico
• Once high-speed entry is complete, the vehicle must begin a parachute deployment sequence (PDS).
  ▪ Jettison Forward Bay Cover (FBC)
  ▪ Deploy Drogue Parachutes
  ▪ Deploy Main Parachutes
Flight Software Triggers

• Typically, triggers are normally simple checks against thresholds:
  
  ```
  if (altitude <= 24000)
    deploy_parachute();
  ```

• When required, additional conditions are added to increase the specificity of the trigger:
  
  ```
  if (altitude <= 24000 && gps_is_available)
    deploy_parachute();
  ```

• When adding new conditions, the number of additional code paths increases, increasing code complexity.

• Code complexity increases software testing costs.

• Therefore, to minimize costs, simple flight software triggers are desired.
Motivation

- By design, deploy drogue parachutes no lower than 24,000 feet altitude, using GPS.

- If both GPS and the backup barometric altimeters are unavailable, then the navigated altitude may have large errors.
Motivation, cont.

• If PDS is still triggered based on navigated altitude in this scenario, many cases deploy too low or too high.

• An alternative trigger must be developed which uses a less error-prone signal.
Introduction to Logistic Regression

• A statistical technique used for fitting a curve to classify data into separate classes.

• Simplest version is binary logistic regression (2 classes of data)

• Fit a logistic function, also known as a sigmoid function (S-curve), to these two classes of data
Logistic Function

\[ f(x) = \frac{1}{1 + e^{-x}} \]
Introduction to Logistic Regression

- The value of the logistic function is bounded between 0 and 1.
- Interpret the value of the logistic function to be the probability that the data $x$ is in class $y = 1$.
- Fit a parameter vector $\theta$ to the training set to minimize model predictive error.
- Once converged, the model is ready to classify new data.

Hypothesis Function

$$P(y|x, \theta) = \frac{1}{1+e^{-\theta^T x}}$$
Logistic Regression Example

Fake Data: Size of Skin Mole vs. Malignancy

Disclaimer: This is not real data. Do not use for medical self-diagnosis.
Logistic Regression Example

Fake Data: Size of Skin Mole vs. Malignancy

Disclaimer: This is not real data. Do not use for medical self-diagnosis.
Applying Logistic Regression to Triggers

- It may be difficult to select a signal to trigger some action in software.

- Logistic regression can aid the designer in determining strong signals for triggers.

- Instead of guessing a trigger signal and tuning a threshold, let an algorithm determine the “best” trigger signals.
Implementation

- Gather training data & labels
  - Training Data: Navigation output from Monte Carlo data
  - Labels:
    - $y=1$ if truth altitude < 25,000 feet
    - $y=0$ if truth altitude $\geq$ 25,000 feet

- Normalize each dimension of the training data to lie within $[-1 \ 1]$ domain (for numerical reasons).

- Fit a logistic function using gradient descent optimization.

- Once converged, the classifier/trigger is ready for use.
Flight Software Implementation

• **Store parameter vector as software parameter.**

```plaintext
function h = logistic(double[] theta, double[] x) {
    double h = 1 / ( 1 + exp(-transpose(theta)*x) );
    return h;
}

function deploy_command = parachute(PARAM, navStates) {
    double h = logistic(PARAM.THETA, navStates);
    boolean deploy_command = h > PARAM.THRESHOLD;
}
```
Performance Analysis

- Identify strong signals
- Combine strong signals
  - Multivariate Logistic Regression
- Performance Criteria:
  - Minimize altitude spread at parachute deployment initiation
  - Ensure minimum deployment altitude stays above 25,000 feet.
Results

• **Selected Navigation Parameters:**
  • Navigated altitude
  • Navigated relative velocity magnitude
  • Elapsed time since sensing 0.2Gs of aerodynamic acceleration
  • Sensed aerodynamic acceleration
  • Navigated Mach number
  • Navigated dynamic pressure

• Record altitude at trigger activation for each trajectory, varying activation thresholds.
Results

Reduced total altitude spread by approximately 3,000 feet as compared to existing approach.
Lessons Learned

- Critical to monitor model fitting process
- May require variable transformations, requiring more designer insight.
- More challenging to understand why a particular trajectory state activated trigger.
- Provides insight into relative value of trigger parameters
Future Work

• Use Naïve Bayes Classifiers to develop triggers
  ▪ Inspired by use in email spam filters

• Preliminary results show large improvements over logistic regression approach.
Conclusion

- Logistic regression is a powerful tool for developing robust flight software triggers.

- Logistic regression can help designer understand the problem space for developing triggers.
An Alternative Flight Software Paradigm:

Applying Multivariate Logistic Regression to Sense Trigger Conditions using Inaccurate or Scarce Information

NASA Johnson Space Center
Kelly Smith, Robert Gay, Susan Stachowiak
AIAA GN&C Conference
Boston, August 20, 2013
Introduction

• Orion is scheduled to fly an orbital test flight, Exploration Flight Test-1 (EFT-1), in late 2014

• Mission Profile:
  o Launch from Kennedy Space Center aboard Delta-IV Heavy into LEO parking orbit
  o After 1 orbit, Delta IV-H upper stage injects the vehicle into an elliptical orbit which intersects the Earth
  o Orion separates from upper stage after apogee
  o Orion flies high-speed guided entry, splashes down in Pacific Ocean west of Mexico
Introduction, cont.

• Once high-speed entry is complete, the vehicle must begin a sequence of parachute deployments.
  o Forward Bay Cover (FBC) Jettison
  o Drogue Parachutes
  o Main Parachutes
Motivation

• By design, the parachute deployment sequence (PDS) is to begin no lower than 24,000 feet altitude.
• When GPS is available, navigated altitude errors are small.
• If GPS is unavailable, navigation altitude will be informed by backup 3 barometric altimeters (less precise).
• If both GPS and the barometric altimeters are unavailable, then the navigated altitude may have large errors.
Motivation, cont.

• If navigated altitude contains large errors, the parachute deployment sequence (PDS) will not occur correctly.
• If PDS is still triggered based on navigated altitude in this scenario, many cases impact Earth before PDS.
• An alternative trigger must be developed which uses a less error-prone signal.
Flight Software Triggers

• Typically, triggers are normally simple checks against thresholds:
  o if (altitude <= 24000) { deploy_parachute(); }

• When required, additional conditions are added to increase the specificity of the trigger:
  o if (altitude <= 24000 && gps_is_available == true) { deploy_parachute(); }

• When adding new conditions, the number of additional code paths increases, increasing code complexity.

• Code complexity increases software testing costs.

• Therefore, to minimize costs, simple flight software triggers are desired.
Introduction to Logistic Regression

• A statistical technique used for fitting a curve to classify data into separate classes.
• Simplest version is binary logistic regression (2 classes of data)
• Fit a logistic function, also known as a sigmoid function (S-curve), to these two classes of data

\[ f(x) = \frac{1}{1 + e^{-x}} \]
Introduction to Logistic Regression

• The value of the logistic function is bounded between 0 and 1.
• Interpret the value of the logistic function to be the probability that the data \( x \) is in class \( y = 1 \).
• Fit a parameter vector \( \theta \) to the training set to minimize model predictive error.
• Once converged, the model is ready to classify new data.

Hypothesis Function

\[
P(y|x, \theta) = \frac{1}{1 + e^{-\theta^T x}}
\]
Logistic Regression Example

Fake Data: Size of Skin Mole vs. Malignancy

Disclaimer: This is not real data. Do not use for medical self-diagnosis.
Logistic Regression Example

Fake Data: Size of Skin Mole vs. Malignancy

Mole Size (millimeters)

Disclaimer: This is not real data. Do not use for medical self-diagnosis.
Applying Logistic Regression to Triggers

- It may be difficult to select a signal to trigger some action in software.

- Logistic regression can aid the designer in determining strong signals for triggers.

- Instead of guessing a trigger signal and tuning a threshold, let an algorithm determine the “best” trigger signals.
Implementation

• Gather training data & labels
  o Training Data: Navigation output from Monte Carlo data
  o Labels:
    • y=1 if truth altitude < 25,000 feet
    • y=0 if truth altitude >= 25,000 feet

• Normalize each dimension of the training data to lie within [-1 1] domain (for numerical reasons).

• Fit a logistic function using gradient descent optimization.

• Once converged, the classifier/trigger is ready for use.
Flight Software Implementation

- Store parameter vector as software parameter.
- Flight software should contain some function to evaluate logistic function:

```plaintext
def logistic(double[] theta, double[] x)
    double h = 1 / (1 + exp(-transpose(theta)*x));
    return h;

function parachute(PARAM, navStates)
    double h = logistic(PARAM.THETA, navStates);
    boolean deploy_command = h > PARAM.THRESHOLD;
```


Performance Analysis

- Identify strong signals

- Combine strong signals
  - Multivariate Logistic Regression

- Performance Criteria:
  - Minimize altitude spread at parachute deployment initiation
  - Ensure minimum deployment altitude stays above 25,000 feet.
Results

- **Selected Navigation Parameters:**
  - Navigated altitude
  - Navigated relative velocity magnitude
  - Elapsed time since sensing 0.2Gs of aerodynamic acceleration
  - Sensed aerodynamic acceleration
  - Navigated Mach number
  - Navigated dynamic pressure

- Record altitude when trigger is first activated for each trajectory for a range of trigger activation thresholds.
Results

Reduced total altitude spread by approximately 3,000 feet as compared to existing approach.
Lessons Learned

• Critical to monitor model fitting process

• May require variable transformations, requiring more designer insight.

• More challenging to understand why a particular trajectory state activated trigger.

• Provides insight into relative value of trigger parameters
Future Work

• Use Naïve Bayes Classifiers to develop triggers
  o Inspired by use in email spam filters

• Preliminary results show large improvements over logistic regression approach.
Conclusion

• Logistic regression is a powerful tool for developing robust flight software triggers.

• Logistic regression can help designers understand the problem space for developing triggers.