

Simple Sensitivity Analysis for Orion GNC

Tom Pressburger¹

NASA Ames Research Center, Moffett Field, California, 94035, USA

Brian Hoelscher²

NASA Johnson Space Center, Houston, Texas, 77058, USA

Rodney Martin³

NASA Ames Research Center, Moffett Field, California, 94035, USA

and

Kumar Sricharan⁴

PARC, Palo Alto, California, USA 94304, USA

The Orion GNC entry team is analyzing the performance of Orion flight software in part by running Monte Carlo simulations of Orion spacecraft flights. The simulated performance is checked for conformance with flight requirements, expressed as performance constraints. Flight requirements include guidance (e.g., touchdown distance from target) and control (e.g., control saturation) as well as performance (e.g., heat load constraints). The Monte Carlo simulations disperse hundreds of simulation input variables, for everything from mass properties to date of launch.

We describe in this paper a sensitivity analysis tool ("Critical Factors Tool" or CFT) developed to find the input variables or pairs of variables which by themselves significantly influence satisfaction of requirements or significantly affect key performance metrics (e.g., touchdown distance from target). Knowing these factors can inform robustness analysis and where engineering resources are most needed, and could even affect operations. The contributions of this paper include the introduction of novel sensitivity measures, such as estimating success probability, and a technique for determining whether pairs of factors are interacting dependently or independently.

The tool found that input variables such as moments, mass, thrust dispersions, and date of launch are significant factors for success of various requirements. Examples are shown in this paper as well as a summary and physics discussion of EFT-1 driving factors that the tool found.

Nomenclature

Cm	= Aerodynamic coefficient of pitching moment
Cn	= Aerodynamic coefficient of yawing moment
CD	= Aerodynamic coefficient of drag
CL	= Aerodynamic coefficient of lift
dCm/dq	= Derivative of aerodynamic coefficient of pitching moment with respect to pitch rate

¹ Computer Scientist, Intelligent Systems Division, MS 269-2.

² Subsystem Manager – Orion Entry Flight Control, Integrated Guidance, Navigation, and Control Branch, MS: EG4

³ Computer Engineer, Intelligent Systems Division, MS 269-1, and AIAA senior member.

⁴ Research Staff, Intelligent Systems Lab, 3333 Coyote Hill Road, Palo Alto, California 94304.

CFT	=	Critical Factors Tool
EFT	=	Exploration Flight Test
FBC	=	Forward Bay Cover (covers the parachutes)
GNC	=	Guidance, Navigation, and Control
L/D	=	Aerodynamic lift over drag ratio
KDE	=	Kernel Density Estimation
MSE	=	Mean Square Error
RBF	=	Radial Basis Function
RCS	=	Reaction Control System
Ycg	=	Y-axis center of gravity offset. The X-axis is the axial centerline of the vehicle.
Zcg	=	Z-axis center of gravity offset.

I. Introduction

THE performance of Orion flight software, especially its GNC software, is being analyzed in part by running Monte Carlo simulations of Orion spacecraft flights. The simulated performance is checked for conformance with flight requirements, expressed as performance constraints. Flight requirements include guidance (e.g., touchdown distance from target) and control (e.g., control saturation) as well as performance (e.g., heat load constraints). The Monte Carlo simulations disperse hundreds of simulation input variables, for everything from mass properties to date of launch (the latter as proxy for weather).

We describe in this paper a sensitivity analysis tool ("Critical Factors Tool" or CFT) developed to find the input variables or pairs of variables which by themselves significantly influence satisfaction of requirements or significantly affect key performance metrics (e.g., touchdown distance from target). Knowing these factors can inform robustness analysis, can inform where engineering resources are most needed, and could even affect operations. The tool produces several measures of the relationship of input variables to both satisfaction of requirements and the key performance metrics. The tool can then sort the input variables by these measures and thus produce a ranking of the independent variables according to their influence on the various requirements and key performance metrics. This paper also describes in Section V preliminary experiments with some regression methods for determining driving factors; otherwise the methods used are rather unsophisticated, relying on deviation of moving averages from the overall mean.

The contributions of this paper include the introduction of novel sensitivity measures, such as estimating success probability, and a technique for determining whether pairs of factors are interacting dependently or independently. The tool found that input variables such as moments, mass, thrust dispersions, and date of launch are significant factors for success of various requirements. Section VI discusses the main driving factors that were found by these techniques for Exploration Flight Test 1 (EFT-1) and their physical understanding. Sections III and IV present the kinds of analysis that the tool performs and will explain each of the figures it generates and how to interpret them. We now discuss related work.

II. Related Work

Sensitivity analysis is used across disciplines wherever models are used, from economics to epidemiology to aerospace. The goal is to find input factors that significantly drive the output of a system. *Factor Mapping*¹ finds input factors that drive the output of the system into specified areas, for example, where requirements are violated. *Monte Carlo Filtering*¹ is a general approach to factor mapping where Monte Carlo simulations are performed and the sets of inputs that drive the output into the specified areas are compared with each other. One suggested sensitivity analysis measure¹ is the expected variance of an output conditioned on knowing the value of an input variable. If knowing the value of the variable reduces the expected variance, then the factor can be considered influential. To efficiently evaluate the conditional variance, special sets of input test vectors are constructed.¹ However, in the Orion setting, we want to make use of the same randomized Monte Carlo input data sets used for GNC analysis. There is a tool set for sensitivity analysis called SimLab² available from the Joint Research Centre of the European Commission at <http://ipsc.jrc.ec.europa.eu/?id=756>; that site has links to other routines as well. A

couple of the techniques we use appear to be implemented in the tool set (Smirnov two-sample¹ and main effect analysis) and future work could experiment with the other tools available there.

Sensitivity analysis has been used in aerospace contexts. The analysis of variance mentioned above was used in an aerospace design context in Ref. 3. Previous work⁴ on sensitivity analysis in an Orion context used the “treatment” learner TAR3^{5,6} to find critical factors. TAR3 finds n -dimensional boxes in the input parameter space in which the desired constraint is particularly success or failure prone. CFT uses TAR3 as well, mostly to confirm the results of CFT’s other techniques. Related work⁷ to Ref. 4 also used a support vector machine to find hyperplanes in the input space that separate successful from failed runs. Other work⁸ in the area of aerospace sensitivity analysis compared estimates of the success and failure probability densities versus an input variable, estimated using Kernel Density Estimation (KDE), to find single influential factors. CFT uses KDE as well (the MATLAB “kde” script⁹ at <http://www.mathworks.com/matlabcentral/fileexchange/14034>).

The KDE method is akin to a method for Monte Carlo filtering that computes the maximum absolute deviation of the success cumulative probability distribution from the failure cumulative probability distribution.¹ This enables application of the two-sample Kolmogorov-Smirnov (two-sided) test¹⁰ for whether the success and failure cumulative probability distributions are statistically different. CFT includes this as well, though we found it is easier to interpret visually the curves for probability densities and success probability than the curves for cumulative probability distribution. The KDE work referenced above⁸ also had techniques for finding higher-dimensional relationships. It used k -nearest neighbors to color, based on the number of nearby successes and failures, a grid versus a pair of factors and then searched for pairs of factors that induced high-contrast regions of success/failure. That work can even find higher than 2-d relationships because the set of factors is expanded to include composite variables, such as the ratio or difference of independent input variables. Instead of k -nearest neighbors, CFT uses the mean absolute deviation, as described in the next section, but does not consider composite variables, whose number can grow quadratically with the number of input variables.

III. Measures of Sensitivity

Each Monte Carlo sample run has recorded with it the success or failure of flight requirements and the values of key performance metrics. As the GNC system matured, the simulations produced fewer requirements failures. To have enough failure data to find input/output success probability sensitivities, the requirements were made more stringent, by strengthening them so that about 10% to 30% of the sample runs failed the more stringent constraint. The assumption is that a trend to failure of the more stringent constraint will carry over to the actual requirement failing. This is not always the case; for an example, see the last paragraph of Section VI and Fig. 6.

The one-dimensional sensitivity measures include: 1) the TAR3 treatment learner’s measure of “lift”; 2) the Pearson product-

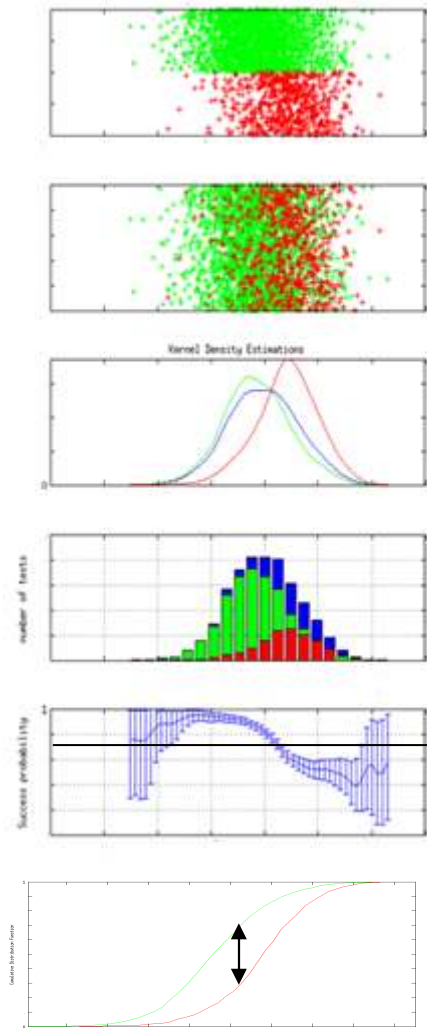


Figure 1. Success/Failure Plots. The X-coordinate is the value of the input variable. The 1st and 2nd plots are scatter plots where the point is colored green if the run satisfied a requirement about touchdown orientation and red if not, and the Y-axis is random. The 3rd plot shows the estimated probability density of all inputs (blue), successful inputs, and failure inputs; the 4th plot is a histogram. The 5th plot shows the success probability confidence intervals as well as a line at the overall mean success probability. A measure of sensitivity is the mean absolute deviation of this line from the success probability confidence intervals. The 6th plot shows the cumulative probability distributions, where the two-sample Kolmogorov-Smirnov test compares against the maximum difference (indicated by the black arrow).

moment correlation coefficient; 3) the average absolute deviation of the success probability density from the failure probability density; 4) the mean absolute deviation of the success probability versus an input variable from the overall mean success probability; 5) the maximum absolute difference between the success and failure cumulative probability distributions (Smirnov); and 6) the mean absolute deviation of the moving average of a key performance metric from its overall mean. The latter is a measure of what has been referred to¹¹ as the *main effect*, $E(Y | X_i = x_i) - E(Y)$; here $E(Y | X_i = x_i)$ is estimated by the moving average. All but the first two of the above measures are “model-free” in that they do not measure goodness of fit to some model, such as an interval or rectangle or linear relationship. On the other hand, discussion of application of regression methods is in Section V.

As mentioned above, one of the tools that CFT employs is TAR3, a “treatment” learner. It finds n -dimensional boxes (in the one-dimensional case, intervals; in the two-dimensional case, rectangles) in the input variable space that are particularly success or failure prone, success meaning relative to meeting the specified flight constraint. TAR3 has its own measure, called “lift”, which is related to how success or failure prone the entire interval is relative to the overall success/failure rate.

We also measure the Pearson product-moment coefficient of correlation of each input variable with success or failure, and the coefficient with each key metric. This coefficient measures the strength of a linear relationship, but can indicate no correlation when in fact there is a U-shaped input/output relationship. It also measures the strength of the linear relationship as opposed to the slope of the relationship.

As mentioned above, the probability densities of the success runs and failure runs as a function of each independent variable are estimated using Kernel Density Estimation and compared; if significantly different, the single input variable has an influence on success. The KDE measure appears to work well no matter how the input variable was dispersed (e.g., from a Gaussian or uniform distribution). An example is shown in Fig 1. The fact that the failure density (in red) is offset to the right of the success density (in green) indicates that higher values of the input variable lead to more failures.

Our new plot shows the success probability of a run satisfying a requirement versus an independent variable (see second to last plot in Fig. 1). This probability can be computed from the success and overall probability density estimates derived using KDE by dividing the success density by the overall density and then multiplying by the overall success probability (for a derivation, see Appendix A). It can also be computed using a moving average by moving a window over the characteristic function that is 1 for success and 0 for failure. Each error bar that is plotted represents a confidence interval around the estimated probability of success, based on how many samples were in the window. It is evident that, at the extreme ends where there are few runs, the confidence interval is large. The confidence interval helps in deciding whether a plot is really showing an effect or one that could be explained by



Figure 2. Success probability.

statistical variation. The intuition for calculating the confidence interval is: suppose there are 100 runs in the window, and 80 are successes, leading to an estimate of 80% probability of success. If that really is the success probability, and if we have a coin that has 80% probability of coming up heads, and we flipped it 100 times, what is the range of number of heads we would come up with 95% of the time (for a 95% confidence interval)? This is a binomial statistics problem. A frequentist approach to finding this confidence interval¹² is to approximate¹² the binomial distribution as a Gaussian distribution with mean p and

standard deviation $\sigma = \sqrt{pq/n}$ where the number of samples n in the window is the number of successes plus the number of failures, p is $\#successes/n$, and $q = 1-p$. The height of the confidence interval is then $2\sigma norminv(0.975, 0, 1)$. The approximation fails, however, when, in the window, there are few samples, or there are no failures.

A Bayesian approach to finding a confidence interval¹³ on the probability computed in a window can be derived by assuming the binomial distribution's conjugate prior, which is the beta distribution, and then updating with the number of success and failure samples in the window. The beta distribution is parametrized by two variables α and β corresponding to the number of successes and failures. It being a conjugate prior to the binomial distribution means that if the prior distribution on the probability is a beta distribution, then the posterior distribution after Bayesian update is also a beta distribution. We use a prior beta distribution where $\alpha_{prior} = p$ and $\beta_{prior} = 1-p$, where p is the overall probability of success. This means that in the absence of other evidence, the estimated probability will tend towards the overall probability of success. The posterior beta distribution for a window has $\alpha_{posterior} = \alpha_{prior} + \#successes$ and $\beta_{posterior} = \beta_{prior} + \#failures$, where $\#successes$ is the number of successes in the window and $\#failures$ is the number of failures within the window. The 95% confidence interval is then derived by the inverse beta distribution function from

$\text{betainv}(0.025, \alpha_{\text{posterior}}, \beta_{\text{posterior}})$ and $\text{betainv}(0.975, \alpha_{\text{posterior}}, \beta_{\text{posterior}})$. The posterior mean μ is $\alpha_{\text{posterior}} / (\alpha_{\text{posterior}} + \beta_{\text{posterior}})$.

Given the confidence interval, one sensitivity measure is the interval's mean absolute deviation from the overall mean success probability, taking into account the confidence interval. More precisely, at each point, we add the distance from the mean success probability to the lower value of the confidence interval when the lower value is above the mean success probability and add the distance to the upper value of the confidence interval when the interval is below the mean success probability. Another measure is the absolute deviation of the overall mean success probability from the success probability weighted by the number of samples in the window used to calculate the success probability. This alleviates a problem in tails of normally-distributed variables of not having sufficient support to justify that the success probability at that point is different from the mean success probability. We found that, compared with comparing probability densities, this success probability method is still somewhat confused by Gaussian-dispersed input variables in that Gaussian-distributed input variables that weren't driving factors would still be ranked highly.

Figure 2 simply maps the success probabilities in Fig. 1 to color, where blue is the highest probability, down to red and then brown for lowest probability. For each such color plot, the overall success probability is mapped to a light green color. If most of the plot is greenish, then there is not much variation from the overall success rate in that variable. The row of colors marked "High" is the color mapping for the tops of the confidence intervals in the success probability figure, and "Low" is for the bottoms of the confidence intervals.

An example of information gleaned from key metrics is in Fig. 3. The X-axis of each mark in the top plot is the value for that run of C_n and in the bottom plot dC_m/dq --clearly both were uniformly dispersed--and the Y-axis is the amount of reaction control system propellant consumed by touchdown. The mark is a red x if the run did not satisfy the more stringent constraint on propellant used. The figure shows the overall mean propellant used, and a moving average of propellant used, calculated using the same windowing approach previously described. We have used a window width of 10% of the range of the input interval (except at the endpoints), sampled at 26 equally spaced locations. Dividing the range into quantiles based on the number of samples and then using the midpoint of the quantile as suggested in Ref. 1, pp. 119 might be better for Gaussian-distributed input variables. It is clear that increasing dC_m/dq leads to on average increased propellant usage, up to the tank capacity. Also computed for each sampled point at which the moving average is calculated is the standard deviation of the samples in the window, which is plotted as a line above and below the moving average. If the lines for the standard deviation have a different slope than the moving average line, then that means that the output dispersion is changing with the value of the variable. It is possible for the moving average line to be horizontal with the dispersion varying, though we have not come across such a situation in practice.

Each one-dimensional sensitivity measure imposes a ranking of variables as to influence, and an overall ranking is produced by intersecting the top-ranked input variables according to the one-dimensional rankings. The tool

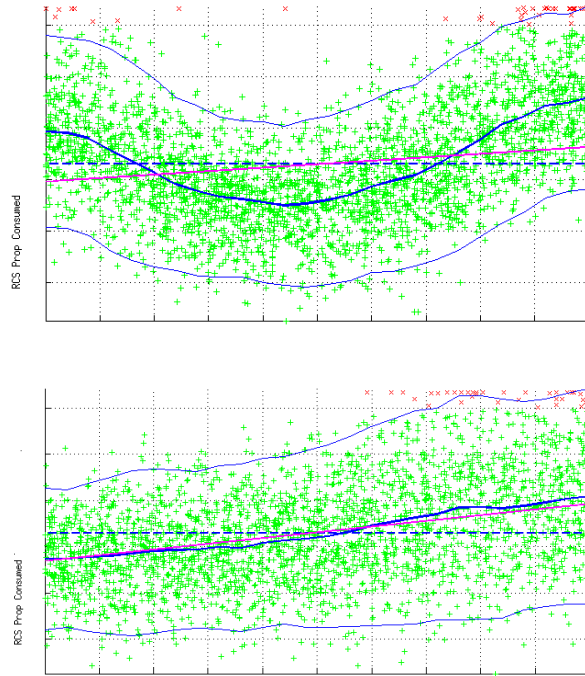


Figure 3. Key Metric Plots. The Y-axis is RCS propellant consumed, and the X-axis is C_n in the top plot and dC_m/dq in the bottom plot. The overall mean and the moving average are shown in blue; the two outer lines are each two standard deviations from the moving average. Runs that violated the propellant-consumed constraint are colored red. The magenta lines were found by the LASSO regression discussed in Section V.

produces an HTML document with the rankings and has links to summary plots containing graphs for the KDE, success probability confidence intervals, and a key metric plot where the latter shows the overall mean, the moving average, and color codes each run as satisfying or not the requirement constraint. At present, the user examines the summary plots in order of rank and develops a subjective impression as to the influence of the input variables.

IV. Two-dimensional Analysis

A two-dimensional sensitivity measure is the mean absolute deviation of the success probability (computed using a moving square window) from the mean success probability. We used a window whose width was 25% of the variable's range (except at the endpoints), sampled by dividing the range into 10 equally spaced intervals. It is important, however, to discern whether the two variables are in fact acting independently, in which case no new information is provided by the 2-d success probability map. We derive in Appendix B a 2-d success probability map based on the product of the two 1-d success probabilities and compare this with the actual 2-d success probability map; this highlights where interactions are occurring. Figure 4 shows an example of 2-d plots for satisfaction of an entry stability requirement where there appears to be little interaction between the variables. The Y-axis is Month, and the X-axis is dCm/dq . From the plots in the 1st row, 1st and 2nd columns (the 1st row, 1st column is oriented vertically because it is for the Y-coordinate), it is clear that both variables have a major effect on whether the requirement was satisfied. The top right plot shows a green “+” where the run is successful and has a Month Y-coordinate and dCm/dq X-coordinate. The larger red squares in the lower right-hand corner of the top right plot indicate that the treatment learner reported that these two variables are interesting together because of the failure prone nature of that area. If we map that figure into colors indicating probability of success, we get the colored square in the 3rd row, 2nd column labeled “Actual Mu”. However, if we look at the 1st row 2nd column scatter plot, we expect the failures to be to the right. If we look at the 1st row, 1st column scatter plot, we expect the failures to be at the top and bottom, and in fact the failures in the plot on the top right are only to the right, top and bottom. One can calculate what the success probability surface would look like if the variables were truly independent (see Appendices B and C), and that gives the colored square in the 4th row, 2nd column labeled “Independent Mu”. It looks about the same as the figure above it, indicating the variables probably don't interact much, though there is a slightly higher probability of success in the middle than might be expected if the variables were truly independent.

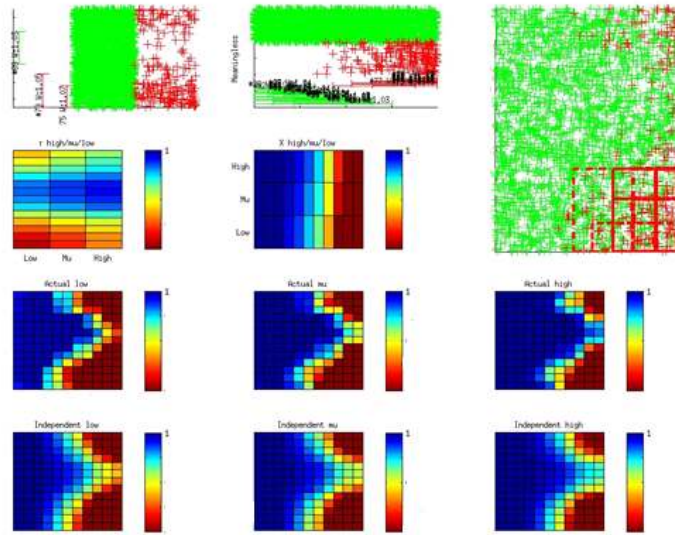


Figure 4. 2-d Independence Analysis. The Y-axis is Month, and the X-axis is dCm/dq . The “low” and “high” plots show the bottom and top, respectively, of the confidence interval around μ .

The Y-axis is Month, and the X-axis is dCm/dq . From the plots in the 1st row, 1st and 2nd columns (the 1st row, 1st column is oriented vertically because it is for the Y-coordinate), it is clear that both variables have a major effect on whether the requirement was satisfied. The top right plot shows a green “+” where the run is successful and has a Month Y-coordinate and dCm/dq X-coordinate. The larger red squares in the lower right-hand corner of the top right plot indicate that the treatment learner reported that these two variables are interesting together because of the failure prone nature of that area. If we map that figure into colors indicating probability of success, we get the colored square in the 3rd row, 2nd column labeled “Actual Mu”. However, if we look at the 1st row 2nd column scatter plot, we expect the failures to be to the right. If we look at the 1st row, 1st column scatter plot, we expect the failures to be at the top and bottom, and in fact the failures in the plot on the top right are only to the right, top and bottom. One can calculate what the success probability surface would look like if the variables were truly independent (see Appendices B and C), and that gives the colored square in the 4th row, 2nd column labeled “Independent Mu”. It looks about the same as the figure above it, indicating the variables probably don't interact much, though there is a slightly higher probability of success in the middle than might be expected if the variables were truly independent.

Figure 5 shows an example where there is a dramatic dependence on a pair of variables of an aerothermal requirement on the backshell. The Y-axis is Cm and the X-axis is Zcg . The plot in the 2nd row, 1st column indicates that there should be fewer failures near the bottom of the 2-d plot, and the plot in the 3rd row, 1st column indicates that not much change should be visible left to right; just a slight improvement at the left and right ends. The “independent mu” plot in the 3rd row 2nd column indicates what would be expected if the variables were truly

independent. Instead, the plot in the 1st row, 1st column and the “Actual” plots show a concentrated diagonal band of failures and a cluster in the lower left.

The Pearson chi-square independence test¹⁰ for categorical variables (that is, variables that take on discrete values) produces a measure of whether two variables are independent. A continuous variable can be made categorical by dividing the variable’s range into intervals. We found that the test was very sensitive to the granularity used to make the variables categorical (i.e., how many intervals into which the range was divided) so the measure turned out to be not very indicative of interesting 2-d relationships.

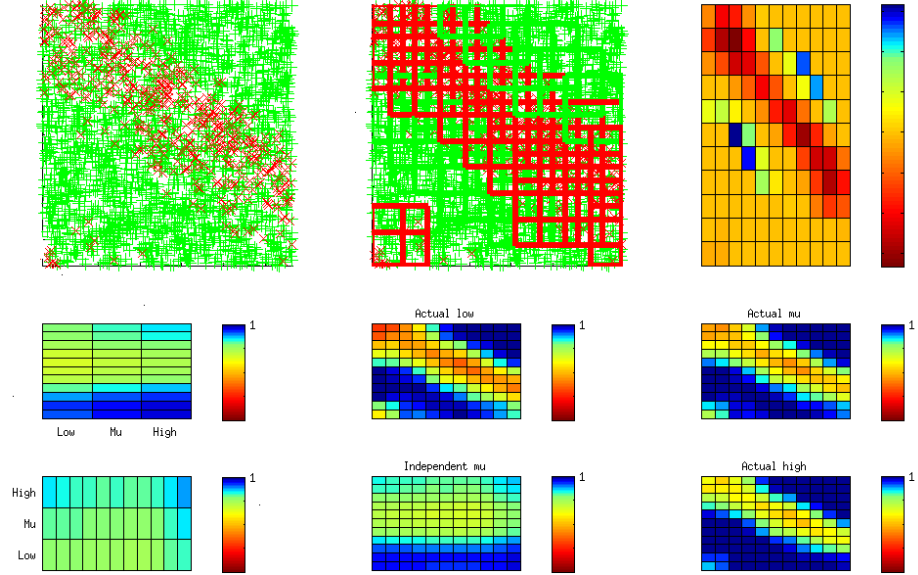


Figure 5. 2-d Independence Analysis. *This is data for an aerothermal constraint. The Y-axis is C_m , and the X-axis is Z_{cg} . Note that “Independent mu”, which is what is predicted if the variables are independent, does not show the band that is shown in Actuals. The top right plot shows reddish squares corresponding to regions where the actual success probability is worse than predicted, and in yellow through blue squares where the actual success probability is better than predicted. The top middle plot shows squares in red for the worse area and squares in green for areas that are better.*

V. Regression Methods

Finally, a variety of regression methods were tested to obtain the best fit to data generated from the Monte Carlo simulations. The original motivation was to use the resulting model as a data-driven proxy; i.e., an equivalent simulation to the physics-based Monte Carlo method at a greatly reduced computational burden; this is called meta-modelling in Ref. 1. However, fitting the Monte Carlo data to a model is useful for sensitivity analysis. For example, a linear fit would yield the coefficients for the input variables; the coefficient’s magnitude (when normalized¹) indicates the sensitivity of the output to the input variable. To facilitate selection of the best performing regression method, the MSE (Mean Square Error) metric was used. Some of the regression methods are parameterized by hyperparameters (for example, the number of hidden units in a single layer perceptron). The MSE metric aided selection of the hyperparameters of the regression methods in order to optimize the MSE as a function of the hyperparameters. One of the regression methods tested was linear ridge regression, in which the associated regularization coefficient was used as a hyperparameter to optimize both the MSE and how numerically well-conditioned the solution is. In statistical learning, a *regularization coefficient* is added to the measure of error of fit to optimize not just the error but other goals as well; for example, the number of non-zero coefficients of a linear fit (see LASSO below).

Regression Method	1 var	2 vars	192 vars
k-nn	0.13974	0.098609	1.0011
LASSO (only linear regressors)	0.14179	0.098783	0.10426 (183 variables selected)
bnet	0.13992	0.096983	0.13469
SVR	0.13984	0.097295	1.0008
Ridge (only linear regressors)	0.14151	0.098823	0.11789
Ridge (including quadratic regressors)	0.13983	0.097329	Did not finish in 5 days.

Table 1. Normalized mean-square errors of the regressions.

regression techniques because it compensates for the difficulty of fitting the data (low variance means the data would be easier to fit). Low values of NMSE indicate better fit.

The main regression method investigated here is a regularized sparse linear regression method known as LASSO (Least Absolute Shrinkage and Selection Operator)¹⁸. Its hyperparameters are already implicitly optimized as part of its algorithm. A valuable benefit which can be derived from using lasso regression is that it can be used to find influential variables due to the nature of its l_1 sparsity “regularization” penalty. LASSO optimizes the sum of the regularization penalty and the measure of the error of the fit. The sparsity penalty penalizes non-zero coefficients associated with the linear regression, meaning that “sparse” solutions will be preferred. Due to the nature of the inequality constraint associated with the regularization penalty, the solution is achieved by appealing to the use of quadratic programming in an iterative fashion, which yields a regularization path of candidate solutions until the algorithm runs to completion based upon the imposition of a convergence or termination criterion. It has been shown¹⁹ that LASSO regression is equivalent to a simple linear correlation analysis, which can be used to select the linearly influential variables as well.

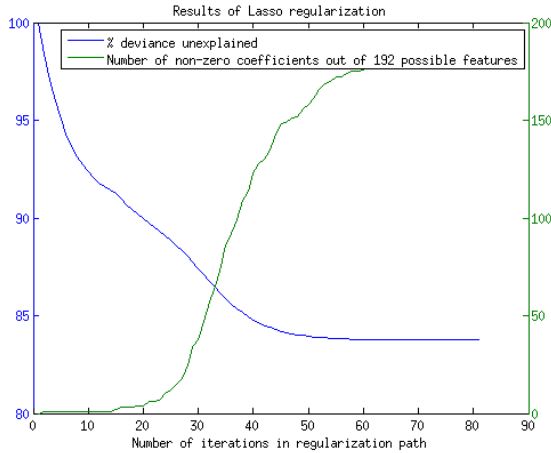


Figure 6. LASSO regularization path.

input variables. During experimental testing, it was found that running the optimization to completion, and choosing the solution at the tail end of the regularization path included most of the variables, and so was not very useful for finding the most influential variables. However, after further experimentation it was found that cutting off the regularization path, based heuristically upon a sufficiency requirement associated with a reasonable approximation

Both linear and quadratic regressors were used, where the use of quadratic regressors include the x_i^2 regressors as well as all $\binom{N}{2}$ quadratic pairs of parameters, $x_i x_j$. Other regression methods included nonlinear methods such as support vector regression (SVR), k-nn (nearest neighbor) regression, and neural networks (bnet). The hyperparameters used for these methods were, respectively, the kernel width of a Gaussian RBF (Radial Basis Function), the number of nearest neighbors, and the number of hidden units in a single layer perceptron. More details on SVR can be found in other work¹⁴ and details on the other methods can be found in Ref. 15 and 16. However, the resulting MSE for all of the methods was not small enough to use as proxy for the physics-based Monte Carlo method, as shown in Table 1 for a total alpha at FBC jettison constraint. The normalized mean-square error¹⁷ (NMSE) is the mean-square error divided by the variance of data; this allows for comparing

Our findings indicate that preselecting the variables to fit based on a linear correlation analysis via the Pearson product-moment correlation coefficient yields reasonable MSE performance, when using the regression methods shown in Table 1, although it is still insufficient for acting as a data-driven proxy to provide an equivalent simulation to the physics-based Monte Carlo method. However, using LASSO on the entire variable set rather than the ones selected with correlation analysis did not cause any substantial degradation in MSE performance of LASSO.

We tested LASSO regression on several different performance requirements, specifically the total alpha at FBC jettison, RCS propellant used, and range to target at touchdown as response variables to be predicted as a function of the 192 independently dispersed simulation

to the percentage deviance unexplained, leads to the same variables found by using the other sensitivity measures such as linear correlation analysis. Figure 6 illustrates this effect by showing the percentage deviance explained and number of coefficients selected by LASSO both as a function of the number of iterations in the regularization path. It is clear that there are diminishing returns beyond a heuristically chosen threshold for a sufficient number of iterations to run the optimization. A precedent for this observation had also been explained in other machine learning work²⁰. In fact, the variables found at 20 iterations were those found by our other techniques. LASSO also returns the coefficients of the linear fit which can be plotted (see Fig. 3). Note that, in the upper plot, it fits a line to a U-shaped relationship.

VI. Results

Monte Carlo data was generated using the Advanced NASA Technology Architecture for Exploration Studies (ANTARES) simulation for Orion EFT-1 vehicle atmospheric re-entry analysis²¹. ANTARES is a six degree of freedom simulation with high fidelity flight dynamics models integrated with the flight software for Guidance, Navigation, and Control functionality. Monte Carlo sets of 3000 cases that each disperse about 200 independent

	Body-rate Induced FBC Jettison	Total Alpha at FBC Jettison	Touch- down Heading Success	Range to Target	RCS Prop Used	Thruster Pulse Count	Instances of Simultaneous Thruster Firing	Bank Satur- ation	Aero- dynamic Load	Backshell Temper- ature
Month	++ O		++ O	++ O	++ O		++ O		++ O	? O
dCm/dq	++	?	?		++		+			
Cm	++ D	++ D		+ D				++	++	++ D
Cn			+ U		++ U	++ U		+	? D	?
CD								?	?	+ D
Mass										++
Ycg offset						?		+	? D	
Zcg offset		++				+	+ D	++ D	++ D	++
Roll jet trail off time							+ D			
Propulsion dispersion			+		?		?			
Roll jet aero interaction uncertainty						+ D				
Parachute twist dispersion			+ D		?	?				
Initial body rates							++ U			

Table 2. Variable Influence. This table shows the influence of variables (rows) on constraints/requirements (columns). Two plus signs (++) means the effect is visually strong; one plus sign (+) means an undeniable but weaker influence; and a question mark (?) means the effect could be questioned. U means U-shaped relationship; D means the metric decreases; O for other.

input variables are used in the analysis discussed here. Each variable was dispersed using either a Gaussian or a uniform distribution.

Table 2 summarizes the influences that dispersed input variables have on various requirements. As mentioned at the end of Section III, we developed a subjective impression of the influence of variables by looking at plots in the order the tool ranked them. Variables with a subjectively strong influence on a requirement are denoted with “++” notation, variables with a undeniable but weaker influence are shown with a “+” notation, and variables with a

possible weak influence are shown with a “?” sign. Thus the tool identifies variables with both a primary influence and a secondary influence. A U-shaped relationship is denoted with a U; a relationship where increasing the variable decreases the metric is denoted by a D; other relationships are designated with an O; and otherwise the metric increases with increasing values of the variable. Prior to using the tool, the Orion Entry team was aware which variables were the primary influence on compliance with particular requirements, but the tool revealed secondary influences that were not known; some are described in the last paragraph of this section.

The Critical Factors Tool identified at least one input variable that is an undeniable influence on compliance with each evaluated requirement metric. The physical understanding of the correlation between the input variables and the required performance will be discussed by grouping the inputs by functional area.

A. Launch date

The atmosphere of the Earth is highly dynamic with significant seasonal variations. The calendar month of the launch date is an input to the simulation atmosphere model and the output characteristics of the atmosphere vary strongly by month. Atmospheric density variation and wind variability are seasonal in nature. The results show that month is the only strong driver for landing range to target. The ability of entry guidance to steer the capsule during hypersonic flight is highly sensitive to density shears in the upper atmosphere. Calendar month is also the only strong influence on touchdown heading accuracy. Orion flight control must point the capsule heading under the main chutes in the direction of the wind velocity at touchdown. Heading control success is highly dependent on wind direction variability. During subsonic flight just prior to parachute deployment, the aerodynamic rate damping is dynamically unstable, so the pitch/yaw body rates are susceptible to wind gusts. When the pitch/yaw body rates reach a certain threshold, the Forward Bay Cover (FBC) jettison command is issued early to deploy parachutes before the vehicle can flip over. Since the level of wind gusts are seasonal in nature, the correlation between launch month and early FBC jettison probability is understood. Greater levels of wind and density variation require the flight control system to work harder to steer and stabilize the vehicle. The strong correlation to propellant usage and instances of simultaneous thruster firings can be understood in this context. Aerodynamic load acceleration is strongly correlated to how hard guidance is working to keep the vehicle on a trajectory that will reach the landing target. This can also be seen in the weak correlation for the bank saturation metric and it is related to the same reason that range to landing target is sensitive to month. The launch date shows weak correlations to aerothermal indicators such as backshell temperature. They are likely also correlated to the variation in guided entry trajectory.

B. Aerodynamics

The Orion capsule atmospheric flight regime encompasses hypersonic aerodynamics to subsonic aerodynamics. The aerodynamic coefficients vary by flight regime. The capability to steer the capsule to a landing target is directly dependent on the level of the lift over drag ratio (L/D). The level of L/D is derived from several factors including the aerodynamics coefficients of lift and drag (CL and CD), the aerodynamics pitching moment coefficient (Cm), and other factors such as mass properties. The dispersion uncertainties on the aero coefficients also vary according to the difficulty in determining the coefficients from wind tunnel testing or computational fluid dynamics analysis. Entry performance will be more sensitive to aero coefficients with larger uncertainties compared to those coefficients that are well known. Each aerodynamic simulation input variable directs the placement of the value of the corresponding aerodynamic coefficient within its uncertainty bands. Several entry performance metrics show strong influence on the pitching moment coefficient Cm. Cm directly affects the trim angle of attack of the capsule which directly affects the L/D ratio. Guidance steering capability is affected by L/D and indicated in the bank saturation, aerodynamic load, and range to target metrics. Aerothermal indicators for the backshell are strongly impacted by trim angle of attack exposing the backshell to the freestream flow. Changes to trim angle of attack also directly impact the total angle of attack at FBC jettison requirement. Because the vehicle has an offset in its center of gravity, the yawing moment coefficient (Cn) acts as a bent airframe inducing body rates into the free bank axis. Flight control must work continuously to counter the bank rates to maintain bank control for landing target steering. This contributes to a strong correlation to total propellant usage and total thruster pulse count metrics. Cn also affects the trim total angle of attack and weaker correlations that are similar to Cm. CD directly affects L/D, though its performance correlations are weaker since its uncertainties are smaller. The aerodynamic rate damping derivative (dCm/dq) becomes unstable during subsonic flight and carries large uncertainties in its measured value. It strongly influences the probability of generating a pitch/yaw oscillation that will trigger an early FBC jettison. It also directly affects propellant usage because of the effort the flight controller puts into actively stabilizing the unstable system during subsonic flight. dCm/dq also weakly correlates to touchdown heading accuracy because

high rates at parachute deployment tend to persist during parachute descent impacting the ability to point the heading at touchdown.

C. Mass Properties

Numerous Orion performance metrics are influenced by vehicle mass properties. Mass has to be constrained to the limits of what the launch vehicle can lift and what the parachute loads can handle. Mass also directly affects aerothermal constraints. A heavier vehicle will experience higher temperatures during atmospheric re-entry. This can be seen from the backshell temperature metrics that are strongly correlated to vehicle mass. The vehicle center of mass directly determines the L/D ratio of the capsule. The vehicle is designed with a deliberately placed Zcg offset with respect to the vehicle centerline to achieve the desired L/D needed to perform the guided entry. Ycg also affects L/D. Though the vehicle is designed with a zero Ycg offset, the Ycg dispersions do impact performance. The Zcg offset affects the vehicle trim angle of attack and has a strong influence on the same metrics discussed in the preceding Aerodynamics section associated with the pitching moment coefficient (C_m). This two dimensional correlation was illustrated earlier in Figure 5 for the effect that Zcg and C_m have on a backshell temperature aerothermal indicator. In addition, the Zcg offset impacts the thruster pulse count and simultaneous thruster firing metrics.

D. Propulsion

The Orion capsule uses 12 thrusters as effectors for entry flight control. The effector model simulates numerous effects such as jet thrust, mass flow rate, aero interactions, thrust build-up/trail-off, engine temperatures, mounting errors, nozzle scarf effects, etc. The model has numerous dispersions. The dispersions that were found to affect entry performance metrics generally were tied to reductions in jet thrust, especially for the roll jets. The roll jets are the work horses that perform bank control during atmospheric flight and heading control under the parachutes. Correlations were found for propellant usage, pulse counts, and simultaneous thruster firings. Touchdown heading accuracy is dependent on roll jet control authority.

E. Parachutes

The Orion system deploys two drogue parachutes followed by three main parachutes for final descent to landing. The fidelity of parachute modeling is critical for analyzing the success of touchdown heading control. One particular dispersion stood out in the results as having a strong influence: the twist dispersion. This parameter directly affects the mean twist torque predicted by the twisting of the main parachute suspension lines. Twist torque subtracts from thruster control authority and therefore directly impacts control of heading orientation. Propellant usage and pulse counts also increase as mean twist torque increases.

F. Initial Body Rates

Orion entry flight phase begins when the capsule separates from the launch vehicle. The dispersion on the initial body rates simulates the range of tipoff rates that can be imparted by the separation mechanism. The flight control algorithm has a bilevel deadband design where one jet is fired for small attitude error signals, but two jets are fired

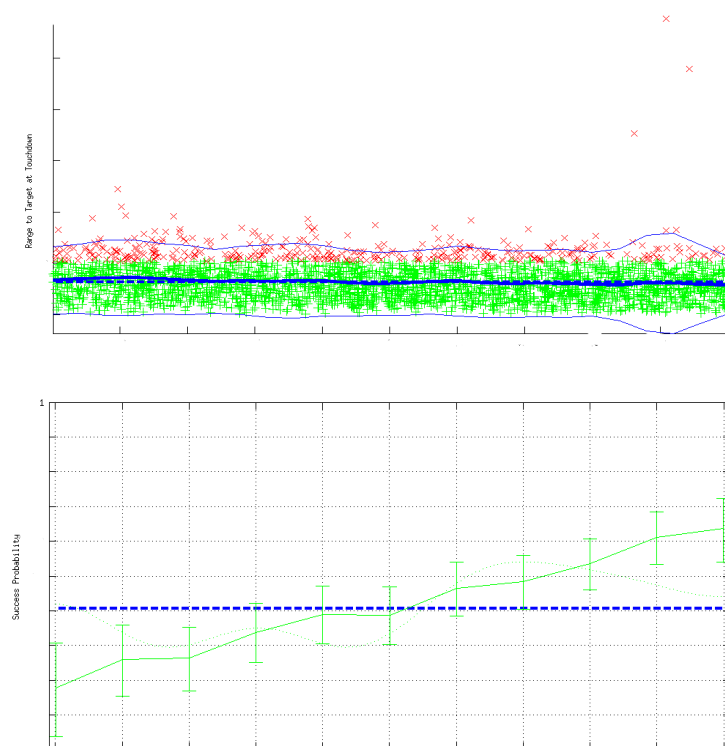


Figure 6. Touchdown Range Constraint. In the top plot, the Y-axis is range to target at touchdown, the X-axis is C_m . The bottom plot shows success probability versus C_m . There are increasingly more failures from right to left, however the out-of-family cases are at the right.

for large attitude error signals. Therefore it comes as no surprise that up to six jets could fire simultaneously after separation for cases where large rates are imparted on the capsule in all three axes.

The Orion GN&C team has found value in the Critical Factors Tool to support its design and analysis tasks. The GN&C team is developing expertise in using the tool for ongoing and future analysis. Previous analysis to identify correlations between input dispersions and key performance metrics were usually based on physical understanding of the problem. The methods were ad hoc and varied by working group team. The Critical Factors Tool formalizes the analysis and allows for identifying primary correlations and secondary correlations. Some of the secondary correlations have been enlightening. For example, the secondary contributions of the RCS jet plume interactions with vehicle aerodynamics led to further studies of GN&C sensitivity to this phenomena. As a direct result of these studies, the Orion aerodynamics team is pursuing additional analysis to refine the jet/aero interaction model used in the simulation tools. Another interesting example is shown in Figure 6 showing the success probability for range-to-target accuracy as a function of the aerodynamic coefficient C_m . GN&C was aware that a high positive dispersion on C_m causes low L/D which can result in landing accuracy failures. The plot shows three cases that are “out-of-family misses” of the landing accuracy cluster at high C_m dispersion. However, the interesting nuance is that the landing footprint shrinks for higher positive C_m dispersions (except for the cases that fail). The explanation is that lower lift reduces the maximum excursions on crossrange as guidance steers the vehicle toward the landing target. It is this kind of subtlety in the statistical output of the Critical Factors Tool that is interesting to Orion GN&C entry analysts. The Critical Factors Tool will be a valuable addition to the entry analysis team’s toolkit for ongoing Orion mission development and design enhancements.

VII. Conclusion

This work shows that simple sensitivity measures can be applied semi-automatically to find critical simulation input variables. This is useful early in the design process, where it is not necessarily known which are the critical factors, as well as later in the design process, where the robustness limits of the designed product are sought.

Appendix

A. Success Probability Derivation

This derives that the success probability at a value x of an independent variable is the ratio of the success probability density to the overall probability density times the overall success probability. S is the success random variable that depends on the value of the input variable. The definitions of the overall success probability and the overall and success probability densities are:

$$\mu = P(S = 1)$$

$$f_{overall}(x) = \lim_{dx \rightarrow 0} \frac{P(t \in [x, x + dx])}{dx}$$

$$f_{success}(x) = \lim_{dx \rightarrow 0} \frac{P(t \in [x, x + dx] | S = 1)}{dx}$$

We wish to calculate

$$P(S = 1 | t = x)$$

$$= \lim_{dx \rightarrow 0} P(S = 1 | t \in [x, x + dx])$$

By Bayes’ rule

$$= \lim_{dx \rightarrow 0} \frac{P(t \in [x, x + dx] | S = 1) P(S = 1)}{P(t \in [x, x + dx])}$$

Dividing numerator and denominator by dx and distributing the limit inwards (assuming they exist and are non-zero) yields

$$= \frac{\lim_{dx \rightarrow 0} \frac{P(t \in [x, x + dx] | S = 1)}{dx} P(S = 1)}{\lim_{dx \rightarrow 0} \frac{P(t \in [x, x + dx])}{dx}}$$

Using the definitions above yields

$$= \frac{f_{success}(x)\mu}{f_{overall}(x)}$$

B. Independence Derivation (1)

This derives the probability map derived from the marginal probabilities assuming two variables are independent.

Let the true 2-d probability map be $p_z(x,y)$. Then the corresponding 1-d marginal versions are the following, where the integrals are defined over appropriate limits. We assume the input variable range is $[0..1]$ so the denominators become 1.

$$p_x(x) = \frac{\int p_z(x,y) dy}{\int dy}$$

$$p_y(y) = \frac{\int p_z(x,y) dx}{\int dx}$$

Now assume $p_z(x,y)$ can be factored into independent probabilities:

$$p_z(x,y) = q_x(x)q_y(y)$$

Define

$$mq_x = \int q_x(x) dx$$

$$mq_y = \int q_y(y) dy$$

Then

$$p_x(x) = \int p_z(x,y) dy = \int q_x(x)q_y(y) dy = q_x(x) mq_y$$

$$p_y(y) = \int p_z(x,y) dx = \int q_x(x)q_y(y) dx = q_y(y) mq_x$$

Hence

$$q_x(x) = \frac{p_x(x)}{mq_x}$$

$$q_y(y) = \frac{p_y(y)}{mq_y}$$

Let the overall average probability be μ . Then

$$\begin{aligned} \mu &= \iint p_z(x,y) dx dy \\ &= \iint \frac{p_x(x)p_y(y)}{mq_x mq_y} dx dy \\ &= \frac{\int p_x(x) dx}{mq_x} \frac{\int p_y(y) dy}{mq_y} \\ &= \frac{\mu}{mq_x} \frac{\mu}{mq_y} \end{aligned}$$

This implies that $mq_xmq_y = \mu$. Together, this implies

$$p_z(x, y) = \frac{p_x(x)p_y(y)}{\mu}$$

C. Independence Derivation (2)

Let $Z(x, y)$ be a random variable with value 0 or 1 indicating failure or success of the system with input x and y . Let N be the number of tests and S be the number of successes. Then $E[Z(x, y)] = S/N = \mu$. We then divide the ranges of x and y into equal sized-bins and estimate

$$\begin{aligned} Q(m) &= E[Z(x, y) \mid x \in xbin_m] \\ R(n) &= E[Z(x, y) \mid y \in ybin_n] \end{aligned}$$

as the following where $S_x(m)$ is the number of successes in the X bin m and $n_x(m)$ is the number of tests in X bin m (similarly, S_y , n_y).

$$\begin{aligned} Q(m) &= \frac{S_x(m)}{n_x(m)} \\ R(n) &= \frac{S_y(n)}{n_y(n)} \end{aligned}$$

We are interested in

$$\begin{aligned} &E[Z(x, y) \mid x \in xbin_m \wedge y \in ybin_n] \\ &= P(Z(x, y) = 1 \mid x \in xbin_m \wedge y \in ybin_n) \end{aligned}$$

because the expected value of a 0,1 function is its probability.

Using Bayes rule in the last equation

$$\frac{P(x \in xbin_m \wedge y \in ybin_n \mid Z(x, y) = 1)P(Z(x, y) = 1)}{P(x \in xbin_m \wedge y \in ybin_n)}$$

The right factor in the numerator can be estimated as μ .

The denominator can be estimated assuming they are independent; if the samples are uniformly distributed then $\frac{n_x(m)}{N} \frac{n_y(n)}{N}$.

The left factor in the result of applying Bayes' rule above, under an assumption of independence, breaks into a product of two factors:

$$P(x \in xbin_m \mid Z(x, y) = 1)P(y \in ybin_n \mid Z(x, y) = 1)$$

Each factor can be estimated directly as the proportion of all successes that fall into the specified bins; e.g., $S_x(m)/S$. Combining our results we get

$$\frac{\left(\frac{S_x(m)}{S}\right)\left(\frac{S_y(n)}{S}\right)\left(\frac{S}{N}\right)}{\left(\frac{n_x(m)}{N}\right)\left(\frac{n_y(n)}{N}\right)}$$

which is $Q(m)R(n)/\mu$.

It can be the case that successes are not independent but failures are. In that case, we can estimate the success rate as

$$1 - \frac{(1 - Q(m))(1 - R(n))}{1 - \mu}$$

As an example, assume $Z(x,y)$ is 1 if $x > 0.7 \wedge y > 0.8$. Then the left factor above is

$$\begin{aligned} & P(x \in xbin_m \wedge y \in ybin_n | Z(x,y) = 1) \\ &= P(x \in xbin_m \wedge y \in ybin_n | x > 0.7 \wedge y > 0.8) \\ &= P(x \in xbin_m | x > 0.7)P(y \in ybin_n | y > 0.8) \end{aligned}$$

If, on the other hand, the constraint was a disjunction like $x > 0.7 \vee y > 0.8$, then factoring into a product above would fail, even though the variables' effect on the condition are "independent" of each other. This is where the failure formula applies:

$$\begin{aligned} & P(x > 0.7 \vee y > 0.8 | x \in xbin_m \wedge y \in ybin_n) \\ &= 1 - P((x \leq 0.7) \wedge (y \leq 0.8) | x \in xbin_m \wedge y \in ybin_n) \\ &= 1 - \frac{(1 - Q(m))(1 - R(n))}{1 - \mu} \end{aligned}$$

Acknowledgments

Tom Pressburger thanks Misty Davies, Steve Fitzgerald, Yuning He, Kurt McCall, Carolina Restrepo and Johann Schumann for discussions about sensitivity/robustness analysis; Michael Lowry, Steve Mueller and Theresa Spaeth for providing Orion support for this work; and Ashok Srivastava and Nikunj Oza for supporting Rodney Martin and Kumar Sricharan.

References

- ¹Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., *Global Sensitivity Analysis: The Primer*, Wiley, West Sussex, England, Feb 2008, Factor mapping and Metamodeling (pp. 156 and Chapter 5), Monte Carlo filtering (pp. 184), Variance-based methods (Chapter 4), Smirnov test (pp. 185), Special test sets (Chapter 4.6).
- ²Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M., *Sensitivity Analysis in Practice*, Wiley, West Sussex, England, 2004. (SimLab instructions).
- ³Jeong, S., Chiba, K., Obayashi, S., "Data Mining for Aerodynamic Design Space," *Journal of Aerospace Computing, Information and Communication*, Vol. 2, Nov 2005, pp. 452-469.
- ⁴Gundy-Burlet, K., Schumann, J., Menzies, T., and Barrett, T., "Parametric Analysis of ANTARES Re-entry Guidance Algorithms Using Advanced Test Generation and Data Analysis," *Proc. i-SAIRAS 2008 (9th International Symposium on Artificial Intelligence, Robotics and Automation in Space)*, 2008.
- ⁵Menzies, T. and Hu, Y., "Just Enough Learning (of Association Rules): the TAR2 "Treatment" Learner," *Artificial Intelligence Review* Vol. 24, 2006, pp. 211-229.
- ⁶Gay, G., Menzies, T., Davies, M., Gundy-Burlet, K., "Automatically Finding the Control Variables for Complex System Behavior", *Automated Software Engineering*, Vol. 17, No. 4, December 2010, pp. 439-468.
- ⁷Barrett, A., "A Parametric Testing Environment for Finding the Operational Envelopes of Simulated Guidance Algorithms," *26th Aerospace Testing Seminar*, Los Angeles, March 29-31, 2011, Institute of Environmental Sciences and Technology, Curran.
- ⁸Restrepo, C. & Hurtado, J. E., "Pattern Recognition for a Flight Dynamics Monte Carlo Simulation," *Proceedings AIAA-Houston Annual Technical Symposium*, Houston, 2010.
- ⁹Botev, Z., Grotowski, J., and Kroese, D. (2010), "Kernel Density Estimation via Diffusion," *Annals of Statistics*, Vol. 38, No. 5, pp. 2916-2957.
- ¹⁰Lindgren, B., *Statistical Theory*, 3rd Edition, MacMillan, New York, 1976, Kolmogorov-Smirnov test (pp. 494), Pearson Chi Square (pp. 424).
- ¹¹Oakley, J. and O'Hagan, A., "Probabilistic Sensitivity Analysis of Complex Models: a Bayesian Approach," *Journal of the Royal Statistical Society: Series B* (2004) Vol. 66, No. 3, , Aug 2004, pp. 751-769.
- ¹²Hoel, P., *Introduction to Mathematical Statistics*, 5th edition, Wiley, 1984. Confidence intervals (pp. 141, Example 2.); Gaussian approximation to binomial distribution (pp. 83-85).
- ¹³Dezfuli, H., Kelly, D., Smith, C. Vedros, K., Galyean, W., *Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis*, NASA/SP-2009-569, June 2009, pp. 29.
- ¹⁴Martin, R., Das, S. "Near Real-Time Optimal Prediction of Adverse Events in Aviation Data," *Proceedings of the AIAA Infotech@Aerospace Conference*, Atlanta, Georgia, April 20-22, 2010.
- ¹⁵Hastie, T., Tibshirani, R., Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition, Springer, 2009.
- ¹⁶Kramer, O. "Dimensionality Reduction by Unsupervised K-Nearest Neighbor Regression," *IEEE 10th International Conference on Machine Learning and Applications*, Honolulu, Hawaii, Dec. 18-21, 2011.

- ¹⁷Bock, J. and Gough, D., “A New Method to Estimate Ligand-Receptor Energetics,” *Molecular and Cellular Proteomics*, Vol. 1, No. 11, Nov. 2002, pp. 906.
- ¹⁸Friedman, J., Hastie, T., and Tibshirani, R. “Regularization Paths for Generalized Linear Models via Coordinate Descent,” *Journal of Statistical Software*, Vol. 33, No. 1, Jan 2010.
- ¹⁹El Ghaoui, L., Viallon, V., Rabbani, T. “Safe Feature Elimination in Sparse Supervised Learning,” EECS Department, University of California, Berkeley Tech Report UCB/EECS-2010-126, Sep. 2010.
- ²⁰Raman, S. “Bayesian Grouped Variable Selection,” Ph.D. Thesis, Computer Science Department, University of Basel, Switzerland, May 2012.
- ²¹Hoelscher, Brian R., “Orion Entry, Descent, & Landing Simulation,” *AIAA Guidance, Navigation, and Control Conference*, August 2007, Hilton Head, South Carolina.