trueSIG user's manual

Sky Sjue
LA-UR 18-23808

1 Introduction

The purpose of this document is to guide potential users of trueSIG, a Python software package for detonator sensitivity testing. The emphasis of trueSIG is to minimize bias of performance estimates with relatively small sample sizes. The nature of the problem is that designers and users of detonators want to be able to assert a very high probability of successful operation with a minimal number of total specimens tested.

Consider a detonator required to function with a 99.9% chance of success. It could be detonated in a variety of ways, but just for example we assume it is a voltage discharged through a very thin wire – an explosive bridge wire. To accurately characterize the 0.1% probability of failure while operating at design conditions and voltage would require more than 1,000,000 trials, because the uncertainty on the probability measured in this type of study is approximately $1/\sqrt{n}$, where n is the number of samples.

To avoid such a prohibitive number of experiments, it is convenient to assume that the detonator responds to the stimulus – the voltage in our example – in a simple and predictible way. We proceed under the assumption that the response can be characterized by one of two models which depend on only two parameters. The models which we will consider are the probit and logit models. In the probit model the probability of success as a function of some stimulus x (possibly voltage) takes the form of a cumulative Gaussian function

$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x} dx' \exp\left(-\frac{(x'-\mu)^2}{2\sigma^2}\right). \tag{1}$$

Similarly, in the logit model the probability is

$$P(x|\mu,\sigma) = \frac{1}{1+\exp[-(x-\mu)/\sigma]}. \tag{2}$$

When $x = \mu$, the probability is 50%. The distance over which the response curves tend to 0 and 1 away from μ is characterized by σ. We will interchangably refer to μ and σ as the location and scale parameters, respectively.

Our software uses Neyer's algorithm [1] with a few important differences which will be explained. The software functions in a GUI with four tabs or pages. We will explain the software's features page by page. An important quantity is the likelihood L for a model given the data, which is defined by the following:

$$L(\mu, \sigma) = \prod_{i=1}^{n} P(x_i|\mu, \sigma)^{y_i}(1 - P(x_i|\mu, \sigma))^{1-y_i}, \tag{3}$$

where we use $y = 1$ for a success and $y = 0$ for a failure. Frequently in calculations we use the log likelihood $\mathcal{L}$ instead of the likelihood,

$$\mathcal{L}(\mu, \sigma) = \sum_{i=1}^{n} \left[y_i \log P(x_i|\mu, \sigma) + (1 - y_i)\log(1 - P(x_i|\mu, \sigma))\right], \tag{4}$$

which is often more convenient as a sum of random variables instead of a product. The starting point is the maximum likelihood estimators (MLEs) $\hat{\mu}$ and $\hat{\sigma}$, which a sufficient set of data uniquely determine via

$$\frac{d\mathcal{L}}{d\mu}(\hat{\mu}, \hat{\sigma}) = \frac{d\mathcal{L}}{d\sigma}(\hat{\mu}, \hat{\sigma}) = 0. \tag{5}$$

The estimate for the 50% threshold $\hat{\mu}$ is unbiased so the asymptotically efficient MLE value is the most ideal possible estimator. However, in general the expectation value for $\hat{\sigma}$ has the form

$$\hat{\sigma} - \sigma = \frac{a_0}{\sqrt{n}} - \frac{a_1}{n} + O_p(n^{-3/2}). \tag{6}$$

In this expression, we have used a_0 and a_1 to signify random variables of order 1. In practice, a_1 can be a factor of a few larger than a_0 and it can take large values of n to obtain results consistent with the central limit theorem. We write the expectation value of the bias as $\mathbb{E}(\hat{\sigma} - \sigma) = b$. The bias and variance both contribute to the mean squared error of the MLE $\hat{\sigma}$, given by

$$\mathbb{E}[(\hat{\sigma} - \sigma)^2] = V[\hat{\sigma}] + b^2, \tag{7}$$

where $V[\hat{\sigma}]$ is the variance of the MLE. The purpose of our modifications to Neyer's algorithm is to remove this bias with fewer data points. This is important when high probabilities of success are required. Reverting to our earlier example of 99.9%, for a cumulative Gaussian response the variance of the probability estimate at the 99.9% level is approximately a factor of 10 more sensitive to σ than μ. In this scenario a d-optimal design is not ideal, so our software defaults to suggesting a c-optimal design with bias reduction. This ensures that the b^2 contribution to the mean squared error in the MLE $\hat{\sigma}$ vanishes with fewer tests than it otherwise would and the confidence will be as high as possible at the probability of interest on the response curve.

The next four sections describe each tab of the GUI software. Screenshots are included in the final section for reference.

2 Configuration

The first page is titled "Configure" and it includes seven parameters. The first three parameters related to μ, along with the guess for σ, guide the suggestions until the data is sufficient to determine unique MLEs. Under the heading "MU - 50% threshold parameter" there are four entries: "best guess," "min guess," "max guess," and "precision." The best guess, max and min for μ take strings to represent floating point values and are used in precisely the same way as in the algorithm diagrammed in Neyer's paper. The parameter labelled precision should be an integer ≥ 0 and it is the number of decimal places which will be carried throughout the software.

Under the heading "SIGMA - length scale parameter" is an entry for "best guess," which is also used in Neyer's algorithm. This entry takes a string which represents a floating point value.

The last heading is "TEST - design specifications" and it is concerned with the intended function of the devices being tested. The entry "design stimulus" takes a string to represent a floating point value. Its potential use in the following pages is to calculate confidence levels as well as the operating margin for a given model. The final entry is "% design probability" and it takes a floating point value to represent a percentage. This value is kept as a variable "dprob," and the software requires and enforces $50 < \text{dprob} < 100$.

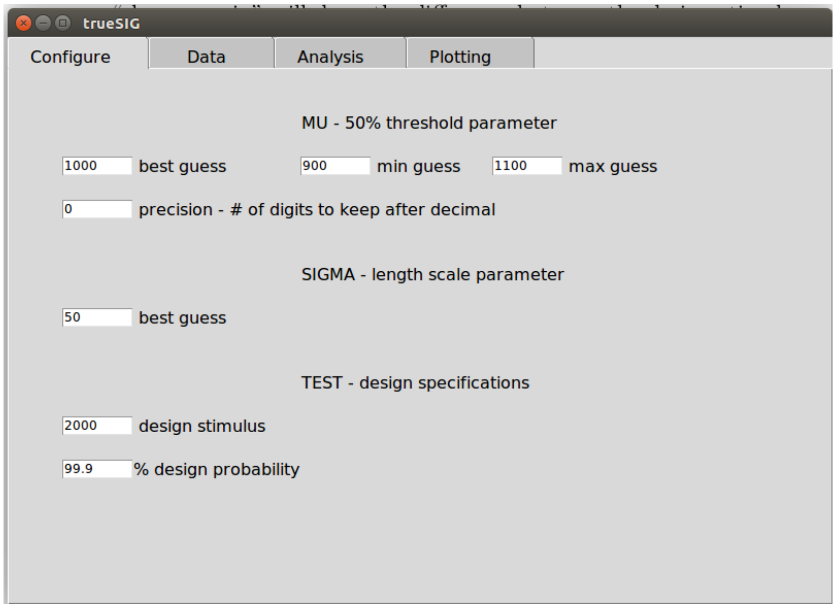

Figure 1: Configure tab with sample settings

3 Data

The second page is titled "Data" and it hosts the data. Everywhere a successful trial will be represented by a value of 1 and an unsuccessful trial or failure will be represented by a value of 0. Files containing data in an ASCII text format can be read using the "Get data from file" button. The file retrieval function is agnostic about file extensions. These files can be text data separated by either white space or commas. A line in a text file will be treated as a comment if it begins with the character #. The following could be an example of a very limited data file:

```
448.0   1
376.0   0
412.0   1
358.0   1
```

The data under consideration is stored in a text box on the data page. This data can be edited by a user in the text box, but data points can also be added on the next page where stimulus values are suggested. There is an

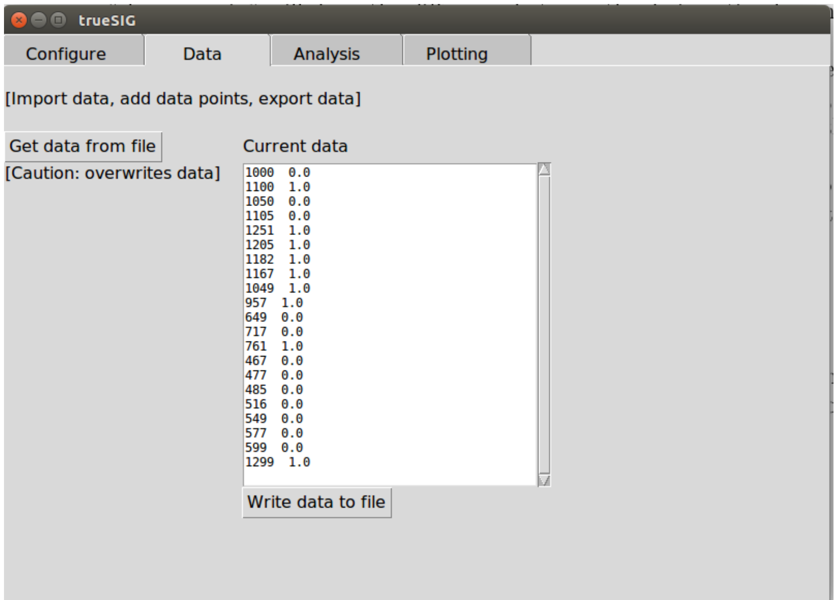

Figure 2: Data tab with some imitation data included

option to save the contents of the text box with the data using a button labelled "Write data to file."

4 Analysis

The next page is labelled "Analysis" and it is intended as the primary interaction point for a user. It has three headings. The first is "Experimental design parameters," which is followed by option menus for model form and design, then finally by a check box to use bias reduction. The model form option menu allows the user to choose from either the logit or probit response curves as described in the Introduction. The design option menu allows a user to choose one of three options. They function as follows:

	logit	probit
d optimal	1.6	1.138
sigma optimal	2.39	1.56
c optimal	<2.39	<1.56

The numbers under logit and probit give the values selected under each design. For example, with a d-optimal design using the logit model, experiments will be suggested at stimuli such that $(x-\mu)/\sigma = \pm 1.6$. The c-optimal option chooses a value such that the asymptotic uncertainty should be minimized at dprob set on the page Configuration. It will always involve values of $(x-\mu)/\sigma$ closer to 0 than the sigma optimal values. As dprob approaches 50%, the values suggested by c optimal will approach $\hat{\mu}$. As dprob approaches 100%, the values suggested by c optimal will approach sigma optimal.

The c optimal value is determined in two steps. The variable dprob defined on the first page (Configure) is in percent, so we define $p_d = \text{dprob}/100$. First we find x_d, the value of x such that $P(x_d|\mu, \sigma) = p_d$. Then we minimize

$$\delta\mu^2(x_c) + \frac{(x_d-\mu)^2}{\sigma^2}\delta\sigma^2(x_c), \tag{8}$$

where $\delta\mu^2(x)$ and $\delta\sigma^2(x)$ are given by the inverse of the Fisher information; they are $\mathbb{E}[-d^2\mathcal{L}/d\mu^2]^{-1}$ and $\mathbb{E}[-d^2\mathcal{L}/d\sigma^2]^{-1}$, respectively. This determines a unique value of $(x_c-\mu)/\sigma$ to be used for a c optimal design, given the probability of interest p_d.

If the option "use bias reduction" is selected the maximum likelihood (ML) estimator $\hat{\sigma}$ will be modified. The ML values $\hat{\sigma}$ and $\hat{\mu}$ maximize $L(\mu, \sigma)$ as defined in the Introduction. The bias reduction option finds two alternative values $\sigma_1 < \hat{\sigma} < \sigma_2$ such that $L(\hat{\mu}, \sigma_1) = L(\hat{\mu}, \sigma_2) = L(\hat{\mu}, \hat{\sigma})/2$. Finally, instead of $\hat{\sigma}$ the software will use $\tilde{\sigma} = (\sigma_1 + \sigma_2)/2$. This procedure reduces the bias in the scale parameter estimator [2]. In fact, for the two binomial response curves described here, we have found this treatment to perform slightly better than other popular methods [3, 4].

The second heading on this page is "Suggestion for next stimulus." It includes an entry giving the suggested stimulus for the next test, which can be modified by the user if the stimulus value in the experiment deviates slightly from the suggestion. There are two buttons, "success" and "failure," which allow the user to add an outcome to the data.

The last heading on this page is "Confidence in design probability at design stimulus." If the user clicks the "calculate" button, two confidence levels will be estimated. The first reads "asymptotic confidence level" and we will refer to it as C_1. It assumes that μ and σ are independent and distributed normally. The expectation value of the Fisher information over the data set is used to get the standard deviations fo these independent normal distributions. The bias reduction option gives a different result for this confidence level since it uses $\tilde{\sigma}$ and the expected Fisher information at this estimate.

To be more precise, C_1 is calculated from two integrals, N_1 and D_1. They take the form

$$N_1 = \left[\int d\mu \int d\sigma \exp\left(-\frac{(\mu-\hat{\mu})^2}{2\delta\mu^2} - \frac{(\sigma-\hat{\sigma})^2}{2\delta\sigma^2}\right)\right]_{P(x_d|\mu,\sigma)\geq p_d} \tag{9}$$

and

$$D_1 = \int d\mu \int d\sigma \exp\left(-\frac{(\mu-\hat{\mu})^2}{2\delta\mu^2} - \frac{(\sigma-\hat{\sigma})^2}{2\delta\sigma^2}\right). \tag{10}$$

The integral for N_1 is constrained such that the response curve has a probability higher than the "design probabiliy" dprob (p_d) at the "design stimulus" x_d. The quantities $\delta\mu^2$ and $\delta\sigma^2$ found from the inverse of the Fisher information. They are the same quantities used to determine the c optimal design. The values of μ are not checked for sign but σ is required to be greater than zero. Finally the estimate for the confidence level is $C_1 = N_1/D_1$.

The second confidence level C_2 uses the observed likelihood. In this case we have analogous quantities

$$N_2 = \left[\int d\mu \int d\sigma L(\mu,\sigma)\right]_{P(x_d|\mu,\sigma)\geq p_d \text{ and } L(\mu,\sigma)>1/2^{n-2}} \tag{11}$$

and

$$D_2 = \left[\int d\mu \int d\sigma L(\mu,\sigma)\right]_{L(\mu,\sigma)>1/2^{n-2}}. \tag{12}$$

These integrals both have an additional constraint. This is necessary because as $\sigma \to \infty$, the likelihood $L(\mu,\sigma) \to 1/2^n$, such that ultimately this integral results in an infinite loop if it is not constrained. The constraint is $L(\mu,\sigma) > 1/2^{n-2}$ since that is consistent with two parameters matching two data points. In this case, the result is similarly $C_2 = N_2/D_2$.

In all cases $C_1 > C_2$. If we refer to the value of C_1 obtained with the bias reduction option selected as C_1', then in general we have $C_1 > C_1' > C_2$. If the data set is inadequate to characterize the response curve to the design probability, then the difference between C_1 and C_2 can be large. An ample set of data can give similar values of C_1 and C_2. The purpose of these dual values is that the user should take notice and consider further studies if $C_2 \ll C_1$ or if C_2 or C_1 are very small in general. In some sense these can be

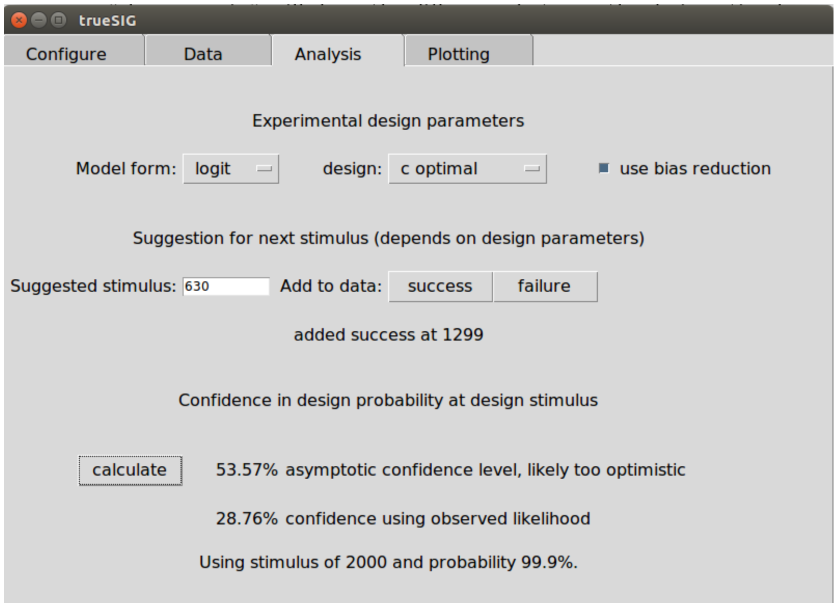

Figure 3: Analysis page with features in use

considered reasonable upper and lower bounds on the confidence level. Even if there is difference between C_1 and C_2, for reasonable sample sizes if the value of C_2 is relatively high the user can be confident in their result.

5 Plotting

The final page facilitates plotting the results. Initially a sinewave is shown. A click on "Refresh" will plot the data. Up to four curves can be drawn with the data at once: ML logit and ML probit, as well as the logit and probit cures using the value $\tilde{\sigma}$ described in the Analysis section. The option "show margin" will draw the difference between the design stimulus and the stimulus at which the response curve reaches the design probability. This requires the inputs to be set on the Configure page. By default the axes will be set by the limits of the data. The axes can be adjusted by hand, then the "freeze axes" option allows a user to replot data and curves with the adjusted axes. Finally, the values of μ, σ and the margin for a given curve can be printed on the plot. These values are printed to the upper left part of

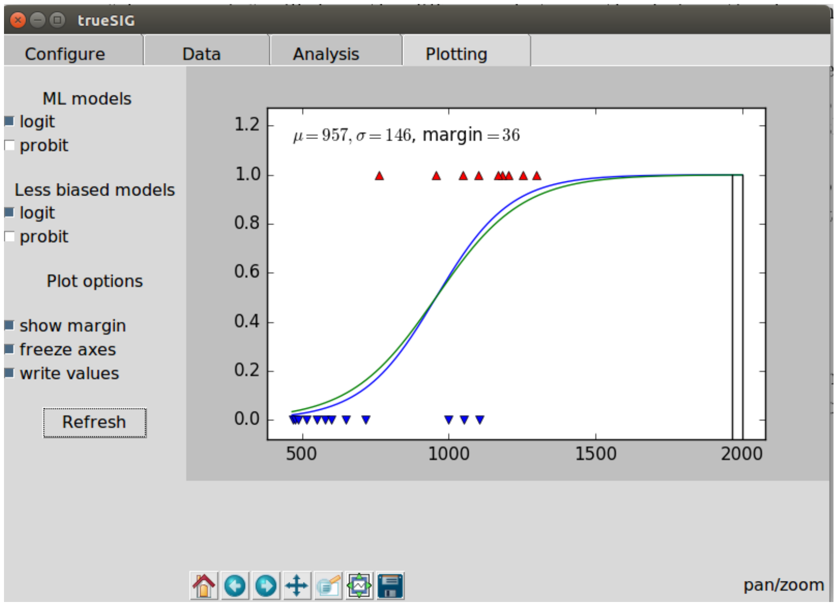

Figure 4: Plots based on imitation data

the plot; readjustment of the axes by hand can achieve a more aesthetically pleasing result.

6 Usage

The software is currently intended to be used in the following way. The files truesig.py and testclass.py should be in a directory and the user should have a prompt/terminal. At this prompt, type (only the part after the $):

```
[sjue@machine ]$python truesig.py
```

Be sure that python refers to the proper installation, which might be best achieved with an alias on Linux or Mac or a .bat file or doskey on Windows. If there is demand in the future preconfigured binary versions could be created for any given operating system. This directory should also contain a file tsconfig.txt if the user desires a modified default configuration on startup.

7 Defaults and modifications

The software can be configured with an input file. At initialization, the software looks for a file called "tsconfig.txt" which supports most of the relevant settings.

Several settings which aren't accessible in the GUI can be changed via tsconfig.txt. The size of the font can be changed. The dimensions of the GUI can be changed. Finally, the number of points used in the integrals to estimate the confidence levels can be adjusted.

The following is an example:

```
#hashtag allows comment
#50% threshold guesses follow
muguess 1000
mugmin 900
mugmax 1100
#significant figures after decimal
precision 0
#guess at scale parameter
sigmaguess 100
#stimulus of interest, design stimulus
designstim 2000
#probability of interest/dprob
designprob 99.9
#designmatrix d, s, or c
#for d optimal, sigma optimal or c optimal
designmatrix c
datafile /path/to/file.ext
#modelform probit or logit
modelform probit
#1/0 for bias reduction/none
reducebias 0
#nclint minimum 2 or greater
#for confidence level integrals
#2 is fastest, higher numbers more precise
nclint 3
#fontsize default is 12
fontsize 14
```

```
#default GUI size is 800x550
guisize 900x600
```

8 Example screenshots

The screenshots included use ubuntu 16.04 LTS with Python 2.7.12. The software has also been tested on Mac OSX Sierra, Red Hat Enterprise Linux 6 with Anaconda Python 3, and Windows 7 with Anaconda Python 3.

References

[1] Barry T. Neyer, *A D-Optimality-Based Sensitivity Test*, Technometrics **36**, 61–70 (1994)

[2] S.K.L. Sjue, D. N. Preston and N. W. Hengartner, *Simplified bias reduction for sensitivity tests*, in preparation for submission to Technometrics

[3] I. Kosmidis and D. Firth, *Bias reduction in exponential family nonlinear models*, Biometrika **96** (4), 793–804 (2009)

[4] I. Kosmidis, *brglm: Bias reduction in binomial-response Generalized Linear Models*, R package version 0.5-9 (2013)