ESTIMATES ON THE COVERAGE OF PARAMETER SPACE USING POPULATIONS OF MODELS

Kevin Burrage
Department of Computer Science
University of Oxford, UK;
Mathematical Sciences School
Queensland University of Technology
Queensland 4001, Australia
kevin.burrage@cs.ox.ac.uk

Pamela M. Burrage
Mathematical Sciences School
Queensland University of Technology
Queensland 4001, Australia
pamela.burrage@qut.edu.au

Diane Donovan
School of Mathematics and Physics
The University of Queensland
Queensland 4072, Australia
dmd@maths.uq.edu.au

Thomas A. McCourt
School of Computing and Mathematics
Plymouth University
Plymouth, UK
thomas.mccourt@plymouth.ac.uk

Harold B. Thompson
School of Mathematics and Physics
The University of Queensland
Queensland 4072, Australia
hbt@maths.uq.edu.au

ABSTRACT
In this paper we provide estimates for the coverage of parameter space when using Latin Hypercube Sampling, which forms the basis of building so-called populations of models. The estimates are obtained using combinatorial counting arguments to determine how many trials, \( k \), are needed in order to obtain specified parameter space coverage for a given value of the discretisation size \( n \). In the case of two dimensions, we show that if the ratio \( (\theta) \) of trials to discretisation size is greater than 1, then as \( n \) becomes moderately large the fractional coverage behaves as \( 1 - e^{-\theta} \). We compare these estimates with simulation results obtained from an implementation of Latin Hypercube Sampling using MATLAB.

KEY WORDS
Populations of models, Latin Hypercube Sampling.

1 Introduction
Mathematical models for describing complex processes often have associated with them a large set of parameters, and some of these parameters may be difficult to estimate. Yet these models are frequently highly tuned with parameters being given to many decimal places. The parameters may be fitted to a set of mean observational/experimental data and the inherent variability in the underlying dynamical processes is often ignored when fitting the parameters. A very recent approach for capturing this important and intrinsic variability is based around the concept of a population of models [1]. In this framework a mathematical model is built that has a set of points in parameter space, rather than a single point in parameter space, all of which are selected to fit a set of experimental/observational data. Such an approach has many advantages in terms of: capturing the intrinsic variability; being able to update the population of models as new data becomes available; and predicting future scenarios from a probabilistic perspective. The disadvantages are the computational aspects in terms of sampling the parameter space in order to calibrate the model against the data, along with analysis and interpretational issues using these probabilistic approaches.

Building populations of models requires the generation of a number of parameter sets for the initial population, sampled from a possibly high-dimensional parameter space. Given a \( d \)-dimensional parameter space, a traditional way of exploring this space has been to choose base values for each parameter and then vary this value by \( 2m + 1 \) (including the base value) percentage changes up and down [2]. Then the total number of parameters to be considered is \( (2m + 1)^d \) and this can become very large as \( d \) increases and/or the resolution, \( m \), is refined. Thus there has been considerable interest in exploring parameter spaces in ways that give good coverage but keep the number of samples to be considered relatively small.

Ideas based on quasi-random draws and, in particular, Halton sequences and digital nets [3, 4, 5] provide sound mathematical bases for sampling techniques such as Latin Hypercube Sampling (LHS). LHS was first described by McKay et al. [6] in 1979. It is a statistical method for generating a sample of plausible collections of parameter values from a multidimensional distribution. Consider a square grid containing one sample in each row and in each column; then a Latin Hypercube is the generalisation of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it. A variant of LHS, known as Orthogonal Sampling, adds the requirement that the entire sample space must be sampled evenly.

There are fundamental implementational issues with respect to random, Latin Hypercube and Orthogonal Sampling. In the case of random sampling new sample points are generated without taking into account the previously generated sample points. On the other hand, in LHS the number of sample points must be determined beforehand and for each sample point the cell in which the sample...
Here we focus on the LHS method. In order to implement the LHS method, upper and lower bounds on the range of values for each parameter are specified and then the range is subdivided into \( n \) equally-spaced sub-intervals. The range is based around mean experimental values. The grid resolution need not be the same for each parameter as the ranges can be different. This procedure generates a subdivision of parameter space into hyper-subcubes. Each subcube is equally likely. Parameter sets are then chosen randomly to satisfy the Latin Hypercube requirements. The number of samples is specified by the user and does not scale with the dimension of the parameter space. Note that the maximum number of combinations for a Latin Hypercube with \( n \) divisions and dimension \( d \) is \((nt)^{d-1}\).

The model-calibration process determines whether or not a particular parameter set should be added to the population, based on a comparison with experimental/observational data. In the cardiac electrophysiological [7, 8] and neuroscience settings [1], biomarkers, such as action potential and beat-to-beat variability, are extracted from time course profiles and then the models are calibrated against these biomarkers. Upper and lower values of each biomarker as observed in the experimental data are used to guarantee that estimates of variability are within biological ranges for any model to be included in the population. If the data cannot be characterised by a set of “biomarkers” then time course profiles can be used and a normalised root-mean-square (NRMS) comparison between the data values and the simulation values at a set of time points can be used to calibrate the population.

In this paper we discuss a method for generating populations of models (where each model is based on a Latin Hypercube) and we provide estimates using combinatorial counting arguments of the expectation of how many trials are needed in order to obtain a specified coverage as a function of the discretisation parameter \( n \). In Section 2 we give some mathematical background to the ideas used in Section 3. In Section 3, we consider the case where the dimension of the parameter space is 2 and briefly discuss how this problem can be posed as a theoretical problem and provide formulae that can be used to estimate the size of the population needed to cover the parameter space. We also confirm these theoretical results by comparing them with a MATLAB implementation of Latin Hypercube Sampling. Section 4 gives some discussion and conclusions.

2 Methods: mathematical background

In 1987, Stein [9] gave the following description of Latin Hypercube Sampling, a description based on ideas presented by McKay, Beckman and Conover and published in 1979 [6]. In what follows we shall assume that \( d = 2 \) so that there are only two parameters in our underlying mathematical model, and that parameter space is subdivided into \( n \times n \) blocks. However, there is a natural generalisation to higher dimensional parameter space. Thus let \( X \) be the set of all possible ordered pairs chosen from \( \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\} \) representing the set of possible parameter values, that are assumed to be independent. So each vector component represents a variable, and its value represents the parameter value, chosen from \( \{1, 2, \ldots, n\} \). We think of the \( n \) values \( 1, 2, \ldots, n \) as dividing each component into \( n \) subintervals and the ranges of each component of \( X \) combine to form \( n^2 \) cells which cover the parameter space. When \( d = 2 \) we may represent a Latin Hypercube Sample of size \( n^2 \) by a \( n \times n \) matrix \( P = [P(j, v)] \), \( 1 \leq j \leq n \) and \( 1 \leq v \leq n \), where each of the columns of \( P \) is an independent permutation of \( \{1, 2, \ldots, n\} \). Each column of the matrix \( P \) is called a trial and, for \( j = 1, \ldots, n \), the notation \( P(j, v) \) represents a model for trial \( v \). Every successful trial in which there are no overlaps with earlier trials adds an additional \( n \) models to the growing population of models. Since the selection of a trial is based on a random permutation, this is a stochastic process and there may be overlaps in a given trial compared to previous trials. If there are \( p \) overlaps in a particular trial then only \( n - p \) models are added to the population of models. Stein [9] discussed the variance between sampling using two techniques (random sampling and Latin Hypercube Sampling) and showed that Latin Hypercube Sampling does reduce the variance relative to simple random sampling. In this paper we focus on the expected coverage of the parameter space as a function of \( k \) trials. We note that there are a number of ways to implement LHS. One approach is to take a larger number for \( n \) and just one trial, so that in this case we can think of \( n \) as being the initial size of our population of models. A different approach is to take a smaller value for \( n \) and a larger number of trials, so that the product of these is the size of the initial population.

By way of an example, if we let \( k = 3 \), \( n = 6 \), then a population of 16 models (as we can see, there are two repeats) that correspond to the Latin Hypercube Sample is given below in terms of a matrix \( P \) made up of columns \( P_1, P_2, P_3 \) and the coverage of a \( 6 \times 6 \) array.

\[
P = \begin{bmatrix}
1 & 3 & 2 \\
2 & 6 & 6 \\
3 & 1 & 4 \\
4 & 4 & 5 \\
5 & 2 & 1 \\
6 & 5 & 3
\end{bmatrix}
\]
3 Results: theoretical and numerical

In this section we focus on Latin Hypercubes with $d = 2$, briefly discussing a theoretical argument that calculates the expected coverage of the parameter space when taking the union of $k$ trials. We achieve this by using combinatorial techniques to count the expected intersection sizes for $m = 2, 3, \ldots, k$ trials chosen from a set of $k$ trials and then use the Principle of Inclusion/Exclusion to count the expected coverage. When $d = 2$, trial $v$ is represented by a column of $P$ and as defined is a permutation of the set $\{1, \ldots, n\}$. So to aid our discussion we exploit this idea, and think of the permutation (trial) as a one-to-one onto function, $f$. That is, given a Latin Hypercube population $P = [P(j, v)]$, $1 \leq j \leq n$ and $1 \leq v \leq k$, for fixed $v$ define

$$f_v : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$$

$$f_v(x) = y, \text{ whenever } P(x, v) = y.$$

There are $n!$ distinct one-to-one and onto functions on a domain of size $n$, and there are

$$\binom{n! + m - 1}{m} = \frac{(n! + m - 1)!}{(n! - 1)!m!}$$

ways to choose (with repetition allowed) $m$ such functions from a set of $n!$ functions.

We say two such functions $f$ and $g$ intersect in $i$ positions if there exists precisely $i$ elements, $x$, of the domain such that $f(x) = g(x)$.

To estimate the number, $U(k, n)$, of cells in the parameter space covered by the union of $k$ trials ($k$ such functions), with $n$ partitions for each of the two parameters, we count via the Principle of Inclusion/Exclusion and calculate the number of cells covered in the parameter space obtaining

$$U(k, n) = \sum_{m=1}^{k} (-1)^m \binom{k}{m} x_m(n), \quad (1)$$

where $x_m(n)$ denotes the expected intersection size of $m$ arbitrary functions. While the formula for $x_m(n)$ becomes increasingly more complex as $m$ increases, we can derive formulae for $m = 1, 2, 3$ and $4$, and we will offer a conjecture on the order behaviour of $x_m(n)$, $m \geq 4$ as a function of $n$.

Trivially $x_1(n) = n$, while a formula for $x_m(n)$, $m = 2, 3, 4$, is obtained by taking the entire set of $n!$ one-to-one and onto functions and then separately considering pairs of functions ($m = 2$), triples of functions ($m = 3$) and quadruples of functions ($m = 4$), that intersect in precisely $i$, $i = 0, 1, \ldots, n$, positions. Summing over all $i$ and dividing by the number of possible choices, with repetition, for $m$ permutations we obtain

$$x_m(n) = \frac{\sum_{i=1}^{n} i z_m(i)}{\binom{n + m - 1}{m}} = \frac{m!(n! - 1)!}{(n! + m - 1)!} \sum_{i=1}^{n} i z_m(i), \quad (2)$$

where $z_m(n) = n!$ and for $0 \leq i \leq n - 1$

$$C_i = \frac{(n!)^2}{i!},$$

$$z_2(i) = C_i \frac{n-i}{2!} \sum_{\alpha=0}^{i} \frac{(-1)^{\alpha}}{\alpha!},$$

$$z_3(i) = C_i \frac{i+2}{3!} \sum_{\beta=0}^{i} \frac{(-1)^{\alpha+\beta}(n-i-\beta)!}{\alpha!\beta!(j-\beta)!} + \frac{2}{3} z_2(i),$$

$$z_4(i) = C_i \frac{i+4}{4!} \sum_{\gamma=0}^{i} \frac{(-1)^{\alpha+\beta+\gamma}(n-i-\gamma)!}{\alpha!\beta!(j-\beta)!\gamma!\alpha!},$$

Here the notation $\sum_{p_j=0}^{L_i}$ denotes a triple summation over the indices $p_j = (j, \beta, \alpha)$ with $L_i = (n-i, j, n-i-j)$, while the notation $\sum_{p_j=0}^{L_5}$ denotes a quintuple summation over the indices $p_5 = (j, \beta, g, \gamma, \alpha)$ with $L_5 = (n-i, j, n-i-j, g, n-i-j-g)$. Furthermore,

$$l_{j,\beta,g,\gamma,\alpha} = \frac{(-1)^{\alpha+\beta+\gamma}(n-i-\beta)(n-i-j-\gamma)!}{(j-\beta)!(j-\beta)!\gamma!\alpha!}.$$
sults.
\[ \sum_{i=1}^{n} i z_2(i) = \frac{n!}{2} (n + n!) \]
\[ x_2(n) = \frac{n(1 + (n - 1)!)}{1 + n!} \]
\[ x_3(n) = \frac{n(1 + (n - 1)! + (n - 1)!^2)}{(1 + n!)(2 + n!)} \]

Based on these results we offer the following conjecture.

**Conjecture:** \( x_m(n) = O\left(\frac{1}{n}\right)^{m-2}, \) \( m = 2, 3, 4, \ldots, \) where the error in this approximation behaves as \( \left(\frac{1}{n}\right)^{m-2} \frac{1}{(n-1)!} \).

Note that this conjecture clearly holds for \( m = 2 \) and \( 3 \) and some further analysis using (2) and the expression for \( z_4(i) \) shows that it is true for \( m = 4 \). Furthermore, we believe that the conjecture is accurate even for modest values of \( n (n > 10) \).

We will use (1) in two ways. First we will construct a truncated approximation, and secondly, we will use the order behaviour in the conjecture to get an accurate approximation to \( U(k, n) \) for values of \( n \) that are moderate or larger. We will find it helpful to let

\[ \theta = \frac{k}{n} \]

and define the percentage coverage to be

\[ P(\theta, n) = \frac{U(k, n)}{n^2}. \]

Then a truncated approximation to \( P(\theta, n) \) after 4 terms is given by

\[ \sum_{j=0}^{3} (-1)^j \prod_{l=0}^{j} \left( \theta - \frac{l}{n} \right) \left( \frac{t_j}{\prod_{l=1}^{j} (l + n!)} \right), \tag{3} \]

where

\[ t_j = \sum_{m=0}^{j} n^m (nl)^{j-m}, \quad j = 0, 1, 2 \]
\[ t_3 = 4n^2 \sum_{i=1}^{n} i z_4(i). \]

We can attempt to estimate the error in (3). We observe that the term in the bracket behaves as \( 1 + \frac{1}{(m-1)!} \). Furthermore, for moderate to larger values of \( n \) we can approximate \( \prod_{l=0}^{j} (\theta - \frac{l}{n}) \) by \( \theta^{j+1} \) and hence a good approximation to the truncation error is \( \theta^{j+1} \frac{\theta^5}{5^j} \). If \( \theta \leq 1 \) this error is acceptable, but if we are trying to estimate the percentage coverage when \( k > n \) then we need a more accurate representation.

This can be achieved by the use of the Conjecture. By applying the order approximations to \( P(\theta, n) \) we get

\[ P(\theta, n) = 1 - \sum_{m=0}^{k} \left( -\frac{1}{n} \right)^m \left( \frac{k}{m} \right) \]
\[ = 1 - \left( 1 - \frac{1}{n} \right)^k \]
\[ = 1 - \left( 1 - \frac{1}{n} \right) \theta n \]
\[ \approx 1 - e^{-\theta}. \tag{4} \]

In the next section we provide estimates of the number of trials to get various percentage coverages as a function of \( n \). These estimates are based on the truncated approximations (3) and the asymptotic approximations (4) and (5).

### 4 Discussions and conclusions

Figures 1 and 2 give the number of trials as a function of \( n \) in order to attain a coverage of 50\% and 70\%, respectively, using the truncated approximation.

![Figure 1. Number of trials to get a coverage of 50\% as a function of \( n \) based on the truncated approximation](image)

Figures 3 and 4 and 5 give the numerical results based on a MATLAB implementation over 100 simulations. We can see that the truncated approximation is only a reasonable estimate when we are looking for a 50\% coverage - see Figures 1 and 3. However, for higher percentage estimates, the truncated approximation is not accurate - see Figures 2 and 4. However, when we examine Figures 3 and 4 and 5, we see that the binomial approximation (4) and the negative exponential approximation (5) compare very well with the numerical estimates of the coverage, even for relatively small values of \( n \) and that as \( n \) become larger these two approximations become increasingly more accurate.

Figure 6 shows how the percentage of coverage asymptotes to 1 as the number of trials increases for a fixed \( n \), in this case \( n = 20 \). Finally, Figure 7 gives estimates of
For each $n$ the minimum number of trials $k$ such that the expected coverage of the parameter space is greater than 70 percent is $5$ subdivisions from 5 to 40.

Figure 2. Number of trials to get a coverage of 70% as a function of $n$ based on the truncated approximation.

Minimum number of trials $k$ such that the expected coverage is greater than 70 percent.

$5$ $0$ $5$ $10$ $15$ $20$ $25$ $30$ $35$ $40$ $45$

Figure 3. Experimental results of trials versus $n$, based on 100 simulations, along with estimates (4, Conjecture 2), and (5, Conjecture 1), to get 50% coverage.

the mean coverage and standard deviation as a function of $S$ simulations for fixed $n = 20$ and $k = 40$. We can see that if the number of simulations is 50 or more, then the mean value is constant, thus suggesting that averaging over 100 simulations is appropriate.

In this paper we have shown how to use sophisticated counting arguments to estimate coverage of parameter space when building populations of models when the number of subdivisions is relatively small and the number of trials is moderate (as opposed to the case when $n$ is the size of the population and there is only one trial). These estimates are accurate and become increasingly more accurate as $n$ increases as shown by comparisons with a MATLAB implementation of Latin Hypercube Sampling. In the case that the ratio of trials to discretisation size, $\theta$, is greater than 1 then as $n$ becomes moderately large the fractional coverage behaves as $1 - e^{-\theta}$. The analysis has been given for a two dimensional parameter space. Similar arguments could, with some additional complexity, be extended to higher dimensional parameter spaces. Of course in these cases the percentage coverage will be much lower in terms of the number of trials and there would be as much interest in the nature of the coverage distribution. This is left for future work in the context of Orthogonal Sampling.

Acknowledgements

The fourth author would like to thank the University of Queensland and the third and fifth authors for their hospitality.

Figure 4. Experimental results of trials versus $n$, based on 100 simulations, along with estimates (4, Conjecture 2), and (5, Conjecture 1), to get 70% coverage.

Figure 5. Experimental results of trials versus $n$, based on 100 simulations, along with estimates (4, Conjecture 2), and (5, Conjecture 1), to get 90% coverage.
Figure 6. Experimental results over 100 simulations to get the relationship between the percentage coverage and the number of trials for a fixed value of $n = 20$.

Figure 7. Estimates of the mean coverage and the standard deviation as a function of $S$ simulations, for fixed $n = 20$ and $k = 40$.

References


