Chemometrics and Intelligent Laboratory Systems 134 (2014) 140–147

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Repairing uniform experimental designs: Detection and/or elimination of clusters, filling gaps ☆

A. Beal, J. Santiago, M. Claeys-Bruno *, M. Sergent

Aix Marseille Université, LISA EA4672, 13397, Marseille Cedex 20, France

ARTICLE INFO

Article history:
Received 26 September 2013
Received in revised form 21 March 2014
Accepted 26 March 2014
Available online 3 April 2014

Keywords:
Experimental designs
Space-Filling Designs
High dimensional space
WSP algorithm
Curvilinear Component Analysis

ABSTRACT

Construction of Space Filling Designs in high dimensional space remains difficult since powerful algorithms at low dimensions become difficult to use at higher dimensions that leads to non-uniform distribution in the factor space. We propose in this paper two approaches in order to repair designs: Curvilinear Component Analysis (CCA) and the Wootton, Sergent, Phan-Tan-Luu's algorithm called WSP in order to detect clusters and to fill gaps. Thus, CCA allows visualization of two or more very closely-spaced points in D dimensions by projecting them in a 2 dimensions space. Then identified clusters can be eliminated using the WSP algorithm. Moreover, the presence of gaps in input space could be very problematic since no information on the phenomenon is available and the WSP algorithm will be used in order to fill gaps by adding points in the "empty" zones. A new quality criterion has been proposed in order to follow the reparation steps. Examples in different dimensions are presented to illustrate these methods.

1. Introduction

In many fields, such as petrochemistry, astronomy, and meteorology, highly complex simulated models are commonly used to represent real phenomena as accurately as possible based on calculation codes. Despite real advances in processor performance, the codes simulating these phenomena still require considerable calculation times. Indeed, increasingly realistic calculations involve a large number of input variables, whose effects can be difficult to predict. It is therefore necessary to develop a strategy to determine the relevant information to supply when producing the model, such as ranking the input variables by order of importance, or having an idea of what the overall phenomenon modeled should look like. This strategy should be as effective as possible and should guarantee good quality information, even at high dimensions. Experimental designs can be used to better organize numerical simulations for this type of approach, and are currently used. However, the number of input variables – often very large (several tens, or even hundreds) – and the wide ranges of variation involved have led to standard experimental designs no longer being really appropriate. This is partly due to how they distribute points (simulations), mainly placing them at the extremities of the variables space. This is why, in numerical simulation, experimental designs known as *Space Filling Designs* (SFD) [1–3], or uniform designs, have become more popular as they distribute the points uniformly throughout the input variables space. However, not all SFD designs are equivalent in terms of the quality criteria reflecting the uniformity of point distribution, such as the intrinsic criteria *Mindist* [4–6] and *Coverage* [7]. *Mindist* is defined as the smallest Euclidian distance between two points. *Coverage* quantifies the homogeneity of spread of points and can be considered as a standard deviation of minimal distances. These criteria allow the comparison of several designs built in the same dimension with the same number of points. The design with the better quality regarding the uniform repartition and the fill-up of the space is characterized by the lowest value of *Coverage* and the highest value of *Mindist*.

In addition, many algorithms which are powerful at low dimensions ($D < 10$) become difficult to use at higher dimensions ($D > 20$ or 30). Thus, low-discrepancy sequences [8–12], such as Faure sequences, present very poor uniformity criteria at high dimensions, with low *Mindist* and high *Coverage* values. The poor conditioning of these experimental designs leads to non-uniform distribution of points throughout the space, causing the appearance of clusters and/or gaps.

Poor conditioning, in terms of non-uniform distribution, can also result from a projection of an experimental design into the sub-space of influential variables revealed by sensitivity analysis. Indeed, after sensitivity analysis, it can be useful to extract the sub-group of factors identified as influential for closer study (modeling) of the phenomenon. This involves keeping the previously performed tests (lines of the design) and only considering the columns representing influential factors. This reduction of the space is known as "folding" and can lead to the appearance of clusters or gaps in the new space.

The aim of this study was to develop a method to repair designs where points are not uniformly distributed throughout the factor space, either because of poor construction or due to folding of the initial

☆ Selected Paper from the 8th Colloquium Chemiometricum Mediterraneum (CCM VIII 2013), Bevagna, Italy, 30th June–4th July 2013.

* Corresponding author. Tel.: +33 491288186.
E-mail address: m.claeys-bruno@univ-amu.fr (M. Claeys-Bruno).

http://dx.doi.org/10.1016/j.chemolab.2014.03.017
0169-7439/

space. To do this, we used Curvilinear Component Analysis (CCA) [13, 14] to visualize clusters. Then designs were repaired using the Wooton, Sergent, Phan-Tan-Luu's selection algorithm (WSP) [15–20] to eliminate any clusters identified and to fill gaps, which strongly penalize the modeling steps. Examples of applications with 2, 8 and 20 dimensions are presented to illustrate these methods.

2. Methods

The methods presented here meet the two objectives presented, i.e., detect the presence of clusters of points in the experimental space and eliminate these clusters if necessary while also filling any gaps. We will present the principles and algorithms for these methods followed by examples of their application.

2.1. WSP algorithm

2.1.1. Algorithm

The WSP algorithm [15–20] allows uniform designs to be rapidly constructed with very good quality criteria, like *Mindist* and *Coverage*. In the WSP selection algorithm, the defined multidimensional parameter space is filled with points selected from a set of candidate points based on a preset minimal distance (d_{min}) from every other point already included in the design.

The algorithm can be summarized as follows:

Step 1 generate a set of N candidate points
Step 2 calculate the distances (D_{ij}) matrix for the N points
Step 3 choose an initial point O and a distance d_{min}
Step 4 eliminate the points I for which: $D_{OI} < d_{min}$. Point O is eliminated from the set of candidate points and will belong to the final subset
Step 5 point O is replaced by the nearest point among the remaining points
Step 6 repeat steps 4 and 5 until there are no more points to choose.

A previous study has shown [18] that the type of the initial candidate design (such as a random design, Latin Hypercubes [21–26], low discrepancy sequences [8–12] and Strauss design [27]) has no importance but only if the number of points is sufficient. The number of candidate points depends on the number of required points in the final design. Santiago et al. [18] advise to consider a number of candidate points equal to at least 5 to 10 times the final set.

Usually the initial point O is chosen as the nearest point of the center of variable space. However, if the candidate design contains a large number of points, whatever the initial point results are identical.

The number of points in the final subset depends on the value of d_{min}. If the d_{min} value increases then the number of points in the final subset decreases. The d_{min} value is determined by iteration until the number of points desired in the final subset is obtained.

Since previous studies [18] have shown that the WSP algorithm leads to uniform designs with good criteria (*Mindist* and *Coverage*) we have chosen to consider this design as presented below.

2.1.2. Reference design

We propose to use a reference design to compare the quality of any designs that could present clusters of points or gaps.

A reference design is constructed with the same dimension and the same number of points to the design to be assessed. The intrinsic uniformity criteria for this design are calculated, and the d_{min} value (equal to the *Mindist* criterion) is used to determine the shortest distance between two points. We then consider that two points separated by a distance shorter than the d_{min} *value* are closer and will form a cluster. If all the points are separated by this d_{min} value, then the spread of points is uniform.

2.1.3. Using the WSP algorithm to detect clusters

Cluster elimination consists in the suppression of points which are closely-spaced in the variable space. It appeared logical to use the WSP selection algorithm for this since this algorithm is based on calculation of distances. The difficulty lies in choosing the d_{min} value which will determine the distance from which a cluster is defined. The d_{min} value will be chosen according to the intrinsic uniformity criteria of a reference design constructed from the same conditions in number of points and dimensions. The *Mindist* is the smallest distance between two points and if we assign this value to the d_{min} then two points separated by a shorter distance than d_{min} are considered as close and will form a cluster.

2.1.4. Using the WSP algorithm to fill gaps

The absence of points in some zones of the space can be problematic as it indicates that no information on the phenomenon is available in this part of the space. The WSP algorithm can be used to fill these gaps. However, this algorithm, which constructs uniform experimental designs, is a selection algorithm retaining a set of points from a set of candidate points. It therefore cannot be used to add points. To overcome this, we concatenated two experimental designs: the one with gaps made up of "protected" points, and a second design containing a very large number of candidate points. The WSP algorithm can then be applied (with a value of d_{min} calculated from the *Mindist* criterion of the reference design) to select points from the sum of these two designs, progressively filling the gaps while retaining the protected points.

2.2. Curvilinear Component Analysis (CCA)

2.2.1. Algorithm

The aim of CCA [13,14] is to reproduce the topology of an initial space of dimension D in a smaller space of dimension p onto which we wish to project all the data. As the overall topology cannot be reproduced, CCA tries to conserve the local topology. To do this, we consider N neurons for which the input vectors $\{x_i;\ i = 1, \ldots, N\}$ in D dimensions quantify the input distribution, and for which the output vectors $\{y_i;\ i = 1, \ldots, N\}$ in p dimensions (where $p < D$) should copy the topology of x_i (Fig. 1). To do this, we use the distances between the x_i: $X_{ij} = d(x_i, x_j)$ where d is the Euclidean distance, and the corresponding output distances are: $Y_{ij} = d(y_i, y_j)$.

During projection, the objective is to make the Y_{ij} distances equivalent to the X_{ij} distances. To do this, we minimize the E_{CCA} criterion (Eq. (1)) characterizing the topological differences between the initial space and the projected space.

$$E_{CCA} = \frac{1}{2}\sum_i \sum_{i \neq j} \left(X_{ij} - Y_{ij}\right)^2 F_\lambda\left(Y_{ij}\right) \tag{1}$$

with $F_\lambda(Y_{ij}) : \mathbb{R}_+ \rightarrow [0, 1]$ a monotone decreasing function of Y_{ij}. This favors local conservation of topology. The $F_\lambda(Y_{ij})$ function is known as the weighting function or function of cost. Demartines and Hérault (1997) [14] first suggested taking function F with parameter λ, known as the critical distance or the neighborhood radius (Fig. 2).

The gradient descent (Eq. (2)) could be used to minimize the E_{CCA} criterion:

$$\Delta y_i = \propto \sum_{j \neq i} \frac{X_{ij} - Y_{ij}}{Y_{ij}} \left[2F_\lambda\left(Y_{ij}\right) - \left(X_{ij} - Y_{ij}\right) F'_\lambda\left(Y_{ij}\right)\right]\left(y_i - y_j\right) \tag{2}$$

with α is the adaptation factor.

However this adaptation rule suffers of several drawbacks. Only one neuron is adapted at a time; thus the adaptation of all neurons is heavy and the adaptation rule can fall into local minima.

Instead of moving one vector y_i according to the sum of contributions of all y_j, the CCA algorithm proposes to fix randomly a point y_i