Robust Experimental Design for Multivariate Generalized Linear Models

Technical Report RP-SOR-0601

Hovav A. Dror and David M. Steinberg

Department of Statistics and Operations Research

Raymond and Beverly Sackler Faculty of Exact Sciences

Tel Aviv University

Ramat Aviv 69978

Israel

Email: dms@post.tau.ac.il

January 2006

Abstract: A simple heuristic is proposed for the construction of robust experimental designs for multivariate generalized linear models. The method is based on clustering a set of local optimal designs, and a method for finding local D-optimal designs using available resources is also introduced. Clustering, with its simplicity and minimal computation needs, is demonstrated to outperform more complex and sophisticated methods.

KEY WORDS: Binary Response; Clustering; Logit; Poisson; Design of Experiments; D-optimal

1. INTRODUCTION

Optimal experimental designs for generalized linear models (GLM) depend on the unknown coefficients, and two experiments having the same model but different coefficient values will typically have different optimal designs. Therefore, unlike experimental design for linear models, the prior knowledge and estimates of the outcome of the experiment must be taken into account. For any given set of values for the model parameters there is an experimental design which is optimal locally. However, since there is uncertainty about the values, one should look for an experimental design that performs well all over the uncertainty space, giving higher priority to regions of higher likelihood within that space.

Prior work on local optimal experimental designs for generalized linear models is mainly focused on a simple linear effect and one design variable, see for instance Abdelbasit and Plackett (1983), Ford, Torsney and Wu (1992) or Mathew and Sinha (2001). Most extensions, for example Sitter and Torsney (1995), are limited to two factors, or to first-order models that do not contain interactions.

Generalizing these results for local optimal designs to take account of uncertainty is even more difficult. Different attitudes toward design robustness for univariate generalized linear models can be found in Abdelbasit and Plackett (1983), Sitter (1992), Hedayat, Yan and Pezzuto (1997) and Chaloner and Larntz (1989). Of these, the latter should be emphasized for suggesting a Bayesian experimental design. Literature concerning multivariate robust designs for GLM is scarce and includes Chipman and Welch (1996), who suggest a minimax

approach, and Robinson and Khuri (2003), who evoke the idea of using so called quantile dispersion graphs. Khuri, Mukherjee, Sinha and Ghosh (2004) survey design issues for generalized linear models. In the survey's conclusion they write "The research on designs for generalized linear models is still very much in developmental stage. Not much work has been accomplished either in terms of theory or in terms of computational methods to evaluate the optimal design when the dimension of the design space is high. The situation when one has several covariates ... demand extensive work to evaluate 'optimal' or at least efficient designs" (Khuri *et al.*, 2004, p. 42). Recently Woods, Lewis, Eccleston and Russell (2005), delivered much of the sought results by proposing a method for finding multivariate compromise designs that allow for uncertainty in the link function, the linear predictor or the model parameters.

In this paper we suggest a simple heuristic capable of finding designs that are robust to most parameters an experimenter might consider, including (similar to Woods *et al.*, 2005) uncertainty in the coefficient values, in the linear predictor equation and in the link function. Its advantages over Bayesian designs such as those of Chaloner and Larntz (1989) or Compromise designs as in Woods *et al.* (2005) are the short computation time required and the simplicity of the method, requiring only the ability to find local optimal designs and a K-means cluster procedure (MacQueen, 1967).

Finding local optimal designs for GLM, and even more so for high-order multivariate models, is far from trivial. Section 2 describes a fast and simple method for finding local D-optimal designs for these complex cases.

Given a set of local D-optimal designs the core of the method proposed is to combine them into a set of location vectors and use K-means clustering to derive a robust design, as motivated by the following examples.

1.1 Example 1

Assume a logistic model with the linear predictor $\eta = \beta_0 + \beta_x x + \beta_y y + \beta_{xy} xy$ having uncertainty about β_0 modeled as a uniform distribution over the region $[0, 2]$ with $\beta_x = \beta_y = 2, \beta_{xy} = 0.2$. Figure 1 shows the local D-optimal designs for this model, for 25 different equally spaced values of β_0 from the feasible region.

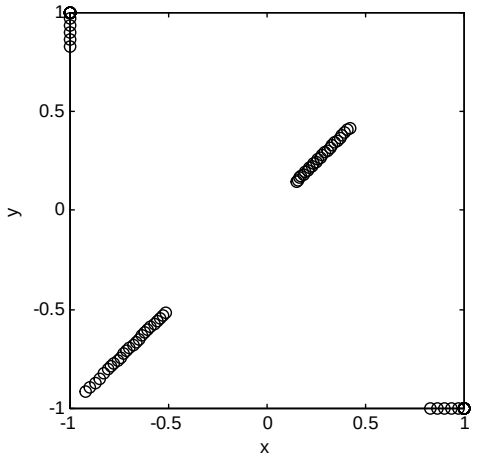

Figure 1: Proximity of 25 local D-optimal designs for a logistic model with intercept value uncertainty

Each local D-optimal design has 4 support points. It can be seen that different values of β_0 result in a small change of the location of these support points, and as a result there is a clear partition of the local designs' points to four clusters. Wishing an efficient experimental design without knowing any further information, it seems reasonable to place one point in each cluster, in its "middle".

1.2 Example 2

Woods *et al.* (2005) noticed that the local D-optimal design for the centroid of the $\beta's$ uncertainty space may often be an efficient compromise design. Preferring the use of clustering can only be justified if it remains an efficient method even in conditions where using the best local D-optimal design fails to perform well.

Continuing example 1 but assuming a larger uncertainty region for β_0 causes the four clusters to overlap. Figure 2 displays local D-optimal designs for 25 different values of β_0 from $[0, 15]$ with $\beta_x = \beta_y = 10, \beta_{xy} = 0.2$. The filled points in the figure show the local D-optimal design for the centroid of the feasible region, $\beta_0 = 7.5$.

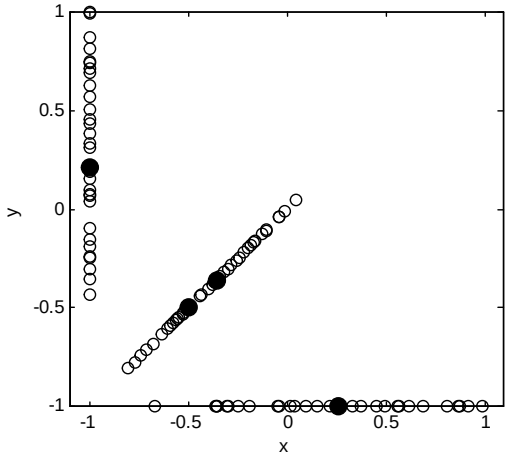

Figure 2: An illustration of the shortcoming of the best local D-optimal design

It is seen that the local D-optimal design for the centroid of the beta space has two support points on the diagonal whose distance from each other is smaller than the range of diagonal point shifts for other possible values of β_0. Coverage of the design space through clustering has better potential for creating a robust design than the parameter space centroid or any other local D-optimal design.

2. FINDING LOCAL D-OPTIMAL DESIGNS

The procedure suggested in this paper assumes the ability to easily construct local optimal designs. The assumption is far from being trivial, as common packages such as "gosset" (Hardin and Sloane 1993), the statistical toolbox in MATLAB (The MathWorks, inc), JMP or the SAS Optex procedure were not designed to be used with Generalized Linear Models.

Finding an exact local D-optimal design for GLM requires finding a choice of n support points that will maximize the determinant of the information matrix. For linear models the information matrix is simply $F'F$, F being the regression matrix. For generalized linear models the information matrix depends on a weights

matrix, and can be represented as $F'WF$ (see for example Atkinson and Donev, 1992). The weights are given by $W = V^{-1}(\boldsymbol{\mu})\left(\frac{d\boldsymbol{\mu}}{d\boldsymbol{\eta}}\right)^2$; V is the variance function, $\boldsymbol{\mu}$ is a vector with row values, μ_i, being the expected response for the experimental configuration expressed by the row F_i of the regression matrix, $\boldsymbol{\eta} = F\boldsymbol{\beta}$ is the linear predictor, $\boldsymbol{\beta}$ is the vector of p unknown coefficients and the relation between μ_i and η_i is expressed through a given link function. For example, for a Poisson model with a log link the diagonal elements of W are $w_{ii} = \mu_i = \exp(F_i\boldsymbol{\beta})$ and for a binary response with the logit link $w_{ii} = \mu_i(1-\mu_i) = \frac{\exp(F_i\boldsymbol{\beta})}{(1+\exp(F_i\boldsymbol{\beta}))^2}$.

Thus, given the values of $\boldsymbol{\beta}$ we can compute the values of the diagonal matrix W for any candidate set of design points. Local D-optimal designs for generalized linear models can therefore be found by setting $\tilde{F} = F\sqrt{W}$ and using a row exchange algorithm, such as Federov's (1972), to find an n point subset of F that maximizes the determinant of the information matrix $\tilde{F}'\tilde{F}$.

For multivariate problems a good candidate set may be of enormous size, causing common computer algorithms to malfunction or preventing their implementation. To overcome this obstacle one may use sequential methods. Begin with a rough grid chosen at random or from a low-discrepancy sequence. For this candidate set calculate the regression matrix and find a D-optimal design. Use the result to create a new candidate set, with each support point of the D-optimal design found being the center of a new random or quasirandom sequence; in order to avoid large candidate sets, limit the size of the sequence around each point so that the number of candidate points from all sequences will be reasonably small; in the examples presented we used 50 normally distributed points around each candidate. Create a rule for adjusting the search radius around the points (for instance reduce the search radius according to the largest distance between points in the new design when compared to the previous step, but no less than 30% of the last search radius used). Create a stopping rule in accordance with the accuracy desired.

For a notion on the effectiveness of the procedure described, it takes less than one second to produce a 16 point local D-optimal design accurate to 2 decimal places for the 5 variable Poisson model containing two interactions used in section 6. Computer run times presented throughout this paper were measured using a desktop PC with a 2.4Ghz Celeron processor.

An implementation of the algorithm, and procedures for examples from the next sections, can be found at http://www.math.tau.ac.il/∼dms/GLM_Design .

3. CLUSTERING VERSUS BAYESIAN DESIGNS

Chaloner and Larntz (1987, 1989) discuss construction of Bayesian optimal designs for a one variable (two parameters) logistic regression where the probability of success for an observation at $x \in [-1,1]$ is $p(x;\beta,\mu) = 1/(1+\exp(-\beta(x-\mu)))$. Their criterion for Bayesian D-optimality is to maximize the average log determinant of the normalized information matrix; the expectation is taken according to a prior distribution of the coefficients (μ,β). Their method requires the number of design points to be specified and so they repeat the optimization (using Nelder and Mead's (1965) simplex algorithm) starting with 2 design points and increasing the number steadily up to 20. They then choose the design that optimizes the criterion on the smallest number of design

points. They illustrate their method with μ and β uniformly distributed on an interval, given three different interval values for each.

They demonstrate that as the uncertainty increases so does the minimum number of support points required to attain the optimal value. But Chaloner and Larntz (1987) also show that out of three intervals examined for μ, only for the widest interval, when it is distributed uniformly on $[-1, 1]$, is the Bayesian design significantly more efficient than the best local D-optimal design. A design based on clustering yields similar results, and has 3 support points for the examples where the Bayesian design has 3 support points. It is more interesting to evaluate the effectiveness of a design based on clustering for the examples in which the Bayesian design proved to be superior to the centroid local D-optimal design, that is for $\mu \sim U[-1, 1]$. As discussed in Chaloner and Larntz (1989) the choice of interval for β has only small influence on the final design and its efficiency, and we will display the results when $\beta \sim U[6, 8]$. Their optimal Bayesian design uses 7 support points with a reported value of -4.5783 for the average log of the information matrix determinant.

We used K-means clustering over 100 local D-optimal designs with the coefficients of β and μ set by a Niederreiter (1988) quasi-random sequence over the described intervals. For a short description of low-discrepancy sequences, see the appendix. Similar to Chaloner and Larntz (1989) the value for K, the number of support points, was increased from 2 to 20. Figure 3 shows the mean value of the log of the determinant matrix when estimated using the same 100 local designs.

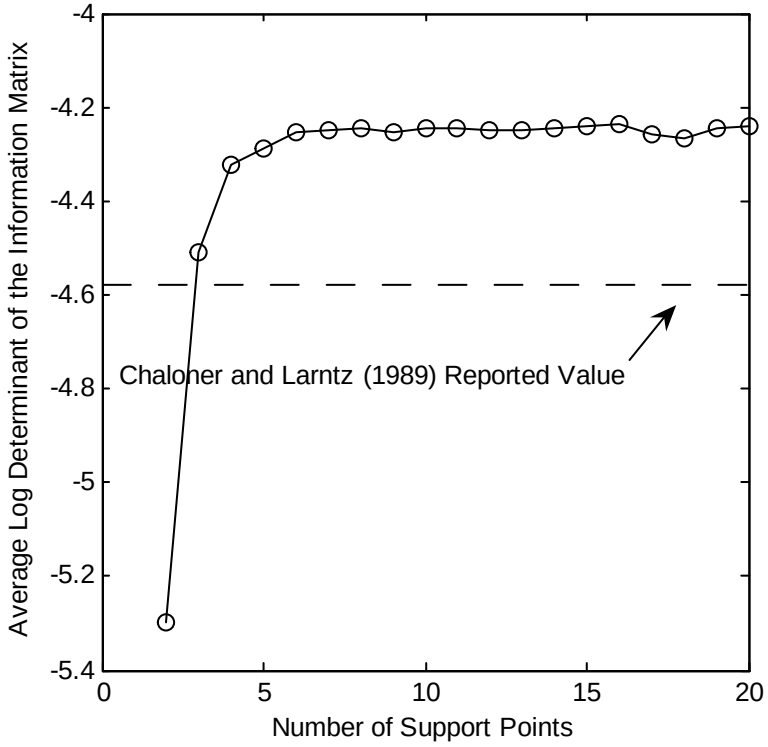

Figure 3: Mean value of the log of the determinant matrix estimated over a rough grid

Similar to the reported result, the criterion value seems to reach a stable value for a design with 7 support points. Its value seems to be better than the one stated by Chaloner and Larntz.

Averaging over 100 designs may be insufficient for a precise evaluation, and the use of the same coefficient values to create the cluster and to estimate its performance may create a bias. We therefore re-evaluated the 7 support point design (created through 100 local D-optimal designs) using 10,000 local D-optimal designs, with their coefficients determined again by a Niederreiter sequence. The criterion value given by this more thorough evaluation approved the validity of the rough estimation. Its value is -4.25, higher than the value reported for the Bayesian design.

One of the advances of Chaloner and Larntz (1989) over former work is to create designs without a requirement for the points to be equally spaced, and with a possibility for a different number of observations at each point. Like their work, a design created by clustering is not restricted to equally spaced points, but it does put equal weight on all the support points. It is possible to adjust the weights and improve the design using sequential quadratic programming on the weights. For the given example this leads to only a minor improvement of the criterion value to -4.23.

Even though creating a robust design using clustering was found superior in this example, one should expect Bayesian designs to be generally better. But, if clustering normally does not fall much from Bayesian designs then it has clear advantages over them - simplicity of creation, and the need for considerably less computational resources. Unlike Bayesian design, extending the clustering procedure to multivariate problems is almost trivial, and is considered next.

4. CLUSTERING VERSUS MULTIVARIATE COMPROMISE DESIGNS

Woods *et al.* (2005) provide a method for finding exact designs for experiments in which there are several explanatory variables. They use simulated annealing to find, like Chaloner and Larntz (1989), a design with a given number of support points that maximizes the average log determinant of the normalized information matrix. They note that evaluating the integral is too computationally intensive for incorporation within a search algorithm, and therefore average over a partial set chosen to represent the model space. Their method allows creation of compromise designs with uncertainty in the link function, the linear predictor and the model parameters.

In section 5, Woods *et al.* (2005) give an example of creating a 16 point compromise design across a parameter space. They describe a crystallography experiment that is aimed to model how four explanatory variables (rate of agitation during mixing, volume of composition, temperature and evaporation rate) affect the probability that a new product is formed. They recommend that when the suggested ranges for the unknowns, β_i, are not large the local D-optimal design for the centroid of the parameter space will be used. Otherwise they find a compromise design based on a coverage design to perform better. The superiority of the compromise design created with the use of a coverage set is demonstrated with a parameter space as described in Table 1 (based on parameter space $\mathcal{B}_3$ in Table 1 of the original paper):

Table 1: Coefficients ranges from Woods *et al.* (2005) crystallography experiment

Parameter	Range
β_0	$[-3, 3]$
β_1	$[4, 10]$
β_2	$[5, 11]$
β_3	$[-6, 0]$
β_4	$[-2.5, 3.5]$

A design's performance was evaluated using the median and minimum efficiency relative to 10,000 local D-optimal designs created for random parameter vectors from the parameter space. The efficiency of a design was calculated as $(|M_C| \,/\, |M_L|)^{1/p}$ where p is the number of unknown coefficients, and M_C and M_L are the information matrices for the evaluated and local D-optimal designs, respectfully. A standard factorial design performed poorly for the example with a median efficiency value of 0.07 and a minimum of 0.003.

Before creating a design using clustering, we examined the compatibility of our assessments to those in Woods *et al.* (2005). We created 10,000 local D-optimal designs using the procedure described in section 2. The values of the 10,000 parameter vectors were produced by a base 2 Niederreiter quasi-random sequence with 2^{12} as a seed. This procedure enables recreation of the exact parameter vectors used here, and at the same time promises a better spread of the parameter space than achieved by random sampling. We then used these designs to evaluate the median and minimum efficiency of the coverage design of Woods *et al.* (2005). The results were compatible with those reported by Woods *et al.*: a median of 0.415 (reported 0.41), and a minimum of 0.113 (slightly lower than the reported 0.12).

The Woods *et al.* (2005) procedure requires their special algorithm and is computer intensive. It is reported by Woods (2005) to require 147 minutes on a stronger computer than we have used. We proceeded, trying to create an alternative design by clustering. The aim of the process was to find a simpler and less computer intensive method for the creation of a design while retaining its robustness.

First we created local D-optimal designs for 100 parameter values, continuing the Niederreiter quasi-random sequence used so far to ensure the use of different local optimal designs for the creation of the composite design and for assessing its efficiency. This preparation work took less than 1 minute. We then gathered the 1,600 resulting points and applied K-means clustering, as implemented in MATLAB (The MathWorks, inc.) to choose 16 representative points as our design. Often optimal design points are found on the boundary of the design region; we therefore used the sum of the absolute differences as a distance measure, so that each cluster is represented by the component-wise median of its points.

Each time clustering is performed, a slightly different design emerges. This is due to the random choice of initial cluster centroid positions. We will therefore summarize design performance via the median (and minimum) efficiencies averaged over 50 identical clustering runs, using the notation *Mean [95% CI]*.

Clustering was found to have competitive results to the Woods *et al.* (2005) composite design, with median

efficiency of 0.40 $[0.38, 0.42]$, and minimum efficiency 0.091 $[0.06, 0.12]$. The time taken to create the composite design (additional to the one minute preparation phase of finding 100 local D-optimal designs) was negligible: 0.25 seconds $[0.16, 0.33]$.

Better results can be obtained by repeating the clustering process numerous times. Similar to Chaloner and Larntz (1989) and Woods *et al.* (2005) we chose the cluster with the highest average log determinant of the information matrix. Averaging was done on the rough grid of 100 parameter vectors that were used to create the local D-optimal designs. Indeed, repeated clustering improved the results: the median efficiency grew to 0.423 $[0.416, .430]$, and the minimum efficiency was 0.096 $[0.06, 0.13]$, requiring only 25 seconds to choose the design.

Furthermore, since clustering is very fast we can easily examine the effect of different choices for the number of support points. Figure 4 displays the result of clustering done with different numbers of support points. At each number of support points we used clustering only once, based on the 100 local D-optimal designs. We approximated the efficiency using the same local optimal designs. Given the local designs, the process of producing the data for the figure took only 20 seconds.

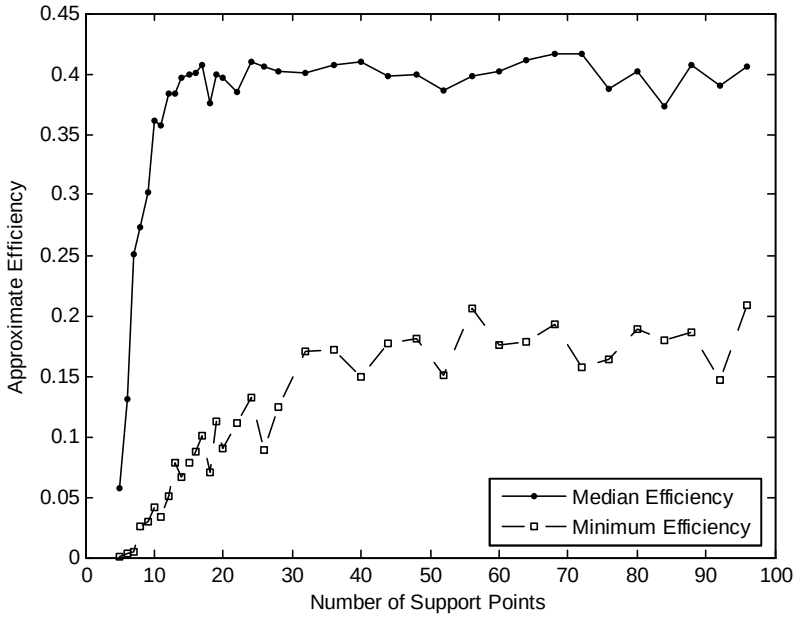

Figure 4: The effect of different choices for the number of support points on the approximated efficiency

From Figure 4 we see that the median efficiency reaches a stable value around the previous number of 16 support points, or slightly above; but, the minimum efficiency continues to grow and only stabilizes for 30 support points or more. We therefore learn that a design with more support points may be advised. In fact, Woods *et al.* (2005) state that in the crystallography experiment 48 observations are to be used, and the 16 point design was to be replicated three times when applied. Other than a 16 point design, Woods (2005) reports the computation time for a 24 point design to be 265 minutes, almost twice as much as their 16-point composite design, which may be the reason for using replicates, rather than considering the option of adding new support

points.

Given this, we chose as before the best design out of 100 repetitions for a 48-means clustering. As expected the median did not change much: 0.423 $[0.415, 0.432]$ but the minimum efficiency increased to 0.177 $[0.141, 0.213]$. The rise in the minimal value is of great importance, as we discuss in section 6. In addition, it is found that efficiency estimation based on 100 local D-optimal designs is quite accurate, so one can produce both a compromise design and an estimate of its efficiency distribution based on a small sample of local designs, which is fast and easy to produce.

Producing the 48 point design, which exceeds in its efficiency Woods' *et al.* (2005) reported results, requires merely an addition of 72 $[63, 80]$ seconds, and combined with the preparation phase take roughly 2% of the time reported by Woods *et al.* (2005).

5. ROBUSTNESS FOR LINEAR PREDICTORS AND LINK FUNCTIONS

Woods' *et al.* (2005) method for finding compromise designs allows uncertainty not only in the model parameters but also in the link function and the choice of the linear predictor. Section 6 of their paper gives an example with two explanatory variables in which there is uncertainty whether a first-order model or a model with the interaction term is more appropriate, and also uncertainty about the link function - Probit versus the asymmetric Complementary-Log-Log (CLL). The values of the model parameters were: $\beta = (3.0, 1.6, 4.1)'$ for the first-order model, and $\beta = (1.2, 1.7, 5.4, -1.7)'$ when considering a model with the interaction term. The results given are for designs with 6 observations.

Woods *et al.* (2005) showed that for this example all of the four local optimal designs perform badly for some of the possible characteristics, with the first-order local D-optimal designs being insufficient for any estimation of the interaction term. A compromise design created for the same problem enables estimation of all four models with efficiencies of at least 0.64. Table 2 is a reproduction of Table 3 from Woods *et al.* (2005), adding a column with the efficiency achieved by clustering the four local D-optimal designs.

Table 2: Efficiencies of a design produced by clustering, Woods *et al.* (2005) compromise design and four local optimal designs d_i, reproduced from Table 3 of the original paper

		Design					
Model		Clustering	Woods	d_3	d_4	d_5	d_6
Probit	No interaction	0.75	0.77	1.00	0.34	0.99	0.30
	Interaction	0.81	0.80	0.00	1.00	0.00	0.97
CLL	No interaction	0.64	0.64	0.99	0.24	1.00	0.11
	Interaction	0.85	0.86	0.00	0.97	0.00	1.00

It is seen that the performance of the Woods *et al.* (2005) compromise design and the design created by clustering the local D-optimal designs is very similar; both achieve at least moderate efficiency for all four models.

In addition to demonstrating the heuristic qualities of clustering, this example is useful to demonstrate a limitation in its usage. Three of the local D-optimal designs included a replicate of the point $[1, -1]$, and so had only 5 support points for a 6 point design. This poses an obstacle for clustering because, while the best design may put higher weight at this support point than on the other design points, the output of the clustering procedure includes any point only once, and if seeking a 6 point design it is likely to replace the replication of the existing point by addition of a different point with inferior contribution. To overcome the obstacle we jittered the points of the local D-optimal designs by a small amount. Indeed, clustering the jittered design points puts two points very close to $[1, -1]$, and is an easy way to overcome the limitation.

6. INK PRODUCTION EXAMPLE

Suppose we have a machine with 5 tubes, each containing a different chemical in a fixed volume, but the concentration of each chemical can be chosen in advance. The machine produces ink which is given a quality classification by the number of imperfect ink marks counted on a standard printed test page. Using low concentrations is assumed to result in low quality ink which is not usable. Although it is expected that the higher the concentrations the higher the quality of the ink produced, high values are preferably avoided, as the concentrations also determine the production cost. An experiment was requested to model the relation between the number of imperfect marks and the concentration of the 5 chemicals.

A Poisson model was used for the quality measure. Prior estimates for the linear predictor and the uncertainty of the model parameters were formed in collaboration with an expert from the factory. The expert was asked to estimate the number of marks for different possible values of the five concentrations; his estimates were analyzed as if they were experimental results, and the uncertainty modeled was approved by the expert as representative. The expert believed a first-order model would be sufficient, but specified two relations between pairs of chemicals to possibly have interaction effects. Both a first-order model and a model with two cross-product terms were constructed from the analysis of the thought experiment, with their results approved by the expert as reasonable representations for both his understanding and his uncertainty of the true model. The estimations are presented in Table 3, for concentration values coded to $[-1, 1]$.

Table 3: Prior coefficients estimates for two models for the ink production example

	First-order		With Interactions	
Term	Estimate	S.E.	Estimate	S.E.
Intercept	-1.52	0.21	-2.35	0.69
x_1	-4.30	0.20	-5.53	0.94
x_2	-1.79	0.16	-2.99	0.82
x_3	-3.39	0.24	-3.95	0.59
x_4	-0.28	0.32	-0.86	0.54
x_5	0.23	0.30	0.41	0.36
x_1x_2			-2.07	1.32
x_1x_3			-1.13	0.98

Remark 1 *Notice that the standard errors are much bigger for the model that contains interactions, even though both models were estimated from the same data. This phenomenon of having less precise estimates for more complex models is common.*

Remark 2 *Our analysis did not take into account the correlation of coefficients, but it can be easily addressed, if desired, in sampling the parameter vectors used to generate local D-optimal designs.*

Estimation of efficiency was done using 20,000 local D-optimal designs - half for the first-order model, and the other half for a model with the suspected interactions. For each model, local D-optimal designs were found for 10,000 coefficient vectors sampled from the normal distribution via a quasi-random sequence.

6.1 Clustering versus a Full Factorial Design

The median efficiency of a full factorial experiment with 32 points is less than 0.1 and, as can be seen in Figure 5, its distribution has 2 peaks, originating from the two models considered (the full factorial has higher efficiency for the first-order model).

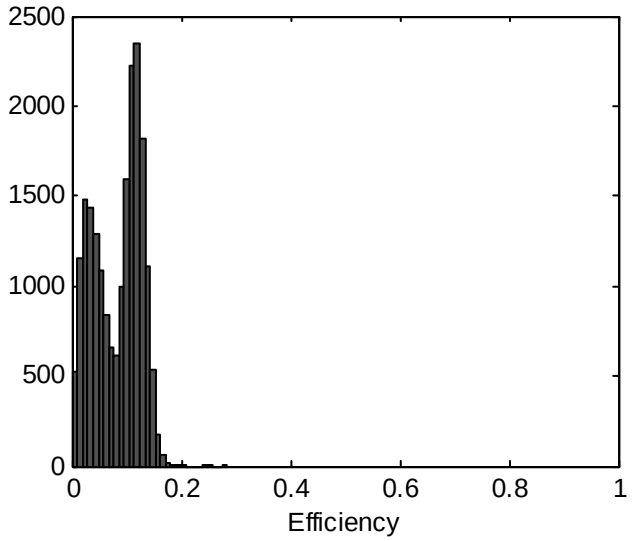

Figure 5: A full factorial design efficiencies histogram for 2 considered models with 10,000 representative model parameters each

The efficiency can be greatly improved by using clustering. As a preparation phase, we created a set of 200 local D-optimal designs, 100 for each model, with parameters taken from a quasi-random sequence, in accordance with the normal distribution assumed. The next step was to choose a good number of support points. We repeated the process used with the crystallography experiment, clustering only once for each of a set of possible support point numbers, and evaluating the efficiency only roughly, over the same set of parameter vectors.

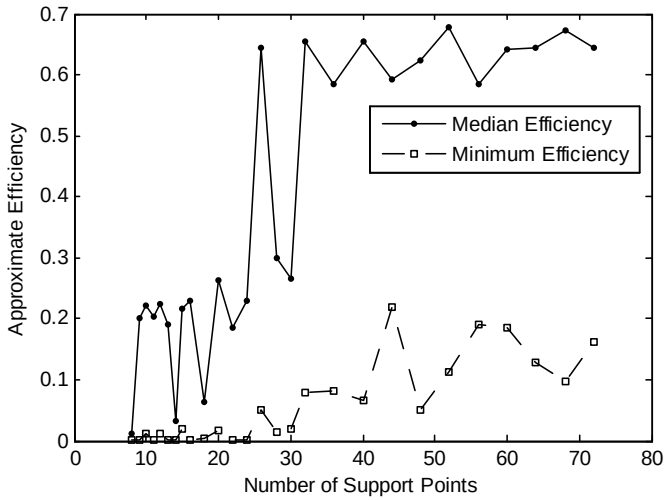

Figure 6: The effect of different choices for the number of support points on the approximated efficiency

Remark 3 *For our purposes it is sufficient to cluster only once, for any tested number of support points, without any repetitions - as was done in the production of Figure 6. But, lack of repetitions causes some of the cases studied to perform very badly, due to a bad random choice of the initial K cluster centroids when*

performing the K-Means clustering procedure. Hence, the graph is not smooth, and the "dips" observed around 15 and 30 support points are likely to be an effect of a poor clustering solution related to the random initial choice of centroids, not to a real problem with these design sizes. Using this graph we choose the desired value for K; then it is important to repeat the clustering process numerous times, to ensure high efficiency.

Remark 4 *It should be assumed that, when the unknown parameters' uncertainty is distributed normally, the true minimum efficiency should approach zero. Hence, the values of the lower curve in Figure 6 are not representative for the minimum values. But, we argue that the lower curve is still a good indicator for the expected change in small efficiency quantiles.*

It is seen that the median efficiency is stable for any choice of more than 30 support points. If we now choose, for example, a design with 48 support points, the median efficiency (as evaluated with the comprehensive database of 20,000 local D-optimal designs) is 0.65; Figure 7 displays a histogram of local efficiencies for a design achieved by clustering.

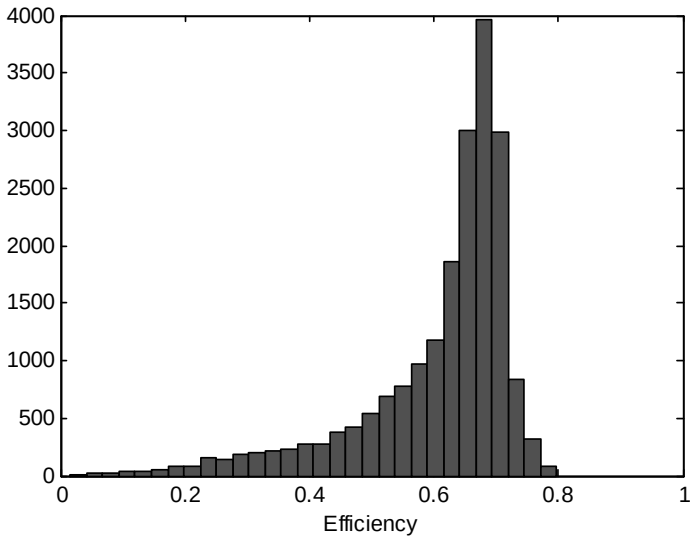

Figure 7: Efficiencies histogram for a 48 point cluster design for 2 considered models with 10,000 representative model parameters each

6.2 Clustering versus Centroid Design

As noted by Woods *et al.* (2005), the local D-optimal design for the centroid of the parameter space is often a sufficiently robust design. When there is more than one model, as in our example, there is no single centroid. Still, having a strong relation between the two examined models, one of the two centroids may be a good choice. Indeed, the local D-optimal design for the richer model is found to perform well, as displayed in its efficiency histogram, Figure 8.

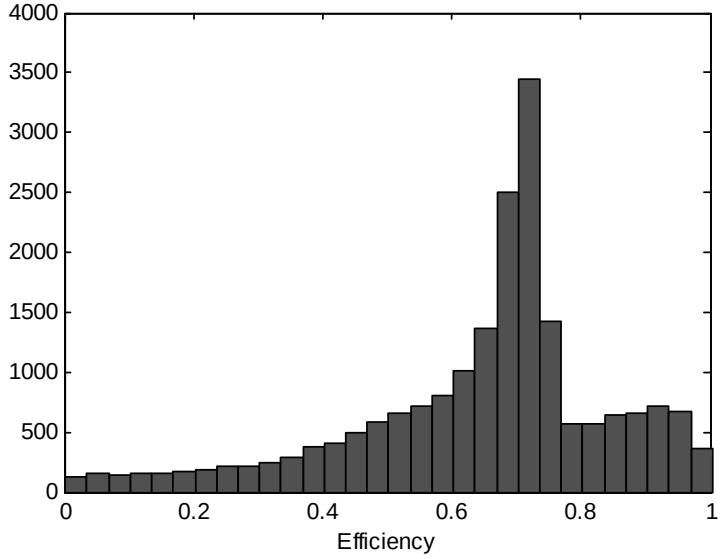

Figure 8: Centroid local D-optimal design efficiencies histogram

The centroid design's median efficiency is even higher in this case than the efficiency achieved by clustering: 0.69. Furthermore, being a local optimal design the histogram is guaranteed to reach a maximum efficiency of 1. As a result, it may seem that a different example would better demonstrate the advantages of creating designs by clustering; possible examples include experiments with more models being considered (perhaps with a larger distinction between them), or a wider uncertainty in the parameter space, as is often the case when the expert cannot give one set of estimates, but considers different scenarios.

But even in this example, the design created by clustering has an advantage over the centroid design, hidden in the left region of the histograms. The relative efficiency between any two designs can be considered as an equivalent sample size; if the relative efficiency of one design is ρ, then it requires $1/\rho$ times as many observations to achieve the same D-criterion value. As is visually obvious (see Figure 9), an efficiency value below 0.2 is related to a drastic increase in the required sample size. It is much more important for a robust design to have as small as possible a fraction of low efficiencies, rather than to include high efficiencies.

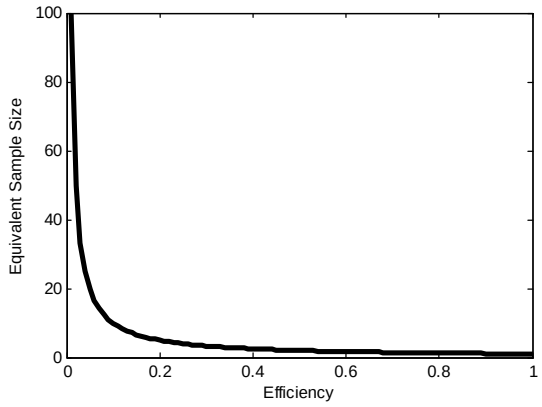

Figure 9: The importance of having as small a portion as possible of low efficiencies

Comparing the efficiency histograms of the centroid and cluster designs, it is seen that the left tail of the

cluster design is thinner. In fact, for the cluster design only 2% of the 20,000 models considered have efficiency smaller than 0.2, in comparison to 4.5% of the models for the Centroid design. Hence, clustering creates a more robust design by decreasing the portion of the uncertainty space that, if discovered to be the true setup, would render the design seriously inefficient.

7. ALGORITHM SUMMARY

We now summarize the algorithm steps for the creation of a robust design through clustering:

1. Translate prior experimental results or experts' opinion into a set of possible models, with their uncertainty defined and estimated.

2. For each model, linear predictor, link function, and/or target criterion, create a sequence of possible parameter vectors, according to a defined distribution, as agreed in the first step. Sampling the parameter space using a low-discrepancy sequence should be preferred over a random sample. In the examples provided, we used 100 vectors produced by a Niederreiter's (1988) low-discrepancy sequence.

3. Find local optimal designs for all the sequences created in step 2 (see section 2 for details).

4. Group the local designs from all models into a single matrix. Apply slight jittering on the components; we decreased from the absolute value of each matrix element a uniformly distributed random variate on $\left[0, 10^{-4}\right]$.

5. Choose a number of support points, K, and use a K-means clustering procedure on the matrix to produce a design. We recommend using the sum of the absolute differences as a distance measure, so that each centroid will be the component-wise median of the points in each cluster (In MATLAB this can be done using the "kmeans" function with the option "cityblock" for distance.)

6. Repeat the process for various choices of K, in order to choose the most appropriate value.

7. For the chosen K value apply clustering numerous times (we used 100 repetitions). After each clustering attempt, calculate the information matrix of the outcome for all the models and parameter vectors chosen in step 2. Sum the log of the determinants of the information matrices. Use the clustering output with the highest sum as your design.

8. CONCLUSIONS

Local D-optimal designs for GLM can be easily found using existing algorithms and computer packages with minor adjustments. Creating a database of local optimal designs in accordance with an a-priori formulation of uncertainty of the model (in the parameter space, the model considered, link function, etc.), can be used to find a design robust to all aspects of the described uncertainty. The heuristic proposed is to then cluster the resulting database. Clearly, this is a simple procedure, requiring minimal computational resources or time even for complex models.

The speed of the process allows exploration of various designs and an investigation of the effect of choosing different numbers of support points is encouraged. Special attention should be paid to finding designs with as small a fraction as possible of very low efficiencies, say lower than 0.2. It has been demonstrated that the ability to explore in a short time many alternative designs helps this simple procedure outperform more sophisticated and complex design optimization methods.

References

1. Abdelbasit, K. M., and Plackett, R. L. (1983), "Experimental Designs for Binary Data," *Journal of the American Statistical Association*, 78, 90-98.

2. Atkinson, A.C. and Donev, A. N. (1992), *Optimum Experimental Designs*, Oxford University Press, Oxford.

3. Chaloner, K. and Larntz, K. (1987), "Optimal Bayesian Design Applied to Logistic Regression Experiments," technical report, University of Minnesota.

4. Chaloner, K. and Larntz, K. (1989), "Optimal Bayesian Design Applied to Logistic Regression Experiments," *Journal of Statistical Planning and Inference*, 21, 191-208.

5. Chipman, H. and Welch, W. (1996), "D-optimal Design for Generalized Linear Models," unpublished manuscript.

6. Federov, V.V. (1972), *Theory of optimal experiments*. New York: Academic Press.

7. Hardin, R. H. and Sloane, N. J. A.(1993), "A New Approach to the Construction of Optimal Designs," *Journal of Statistical Planning and Inference*, 37 339-369.

8. Hedayat, A.S., Yan B. and Pezzuto, J.M. (1997), "Modeling and Identifying Optimum Designs for Fitting Dose-Response Curves Based on Raw Optical Density Data," *Journal of the American Statistical Association*, 92, 1132-1140.

9. Khuri, A.I., Mukherjee, B., Sinha, B. and Ghosh, M. (2004), "Design Issues for Generalized Linear Models," Technical Report No. 2004-016, Department of Statistics, University of Florida, Gainesville, FL 32611.

10. MacQueen, J. B. (1967), "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability," Berkeley, University of California Press, 1, 281-297.

11. Nelder, J.A. and Mead R. (1965), "A Simplex Method for Function Minimization," *Computer Journal*, 7, 308-313.

12. Niederreiter, H. (1988), "Low-Discrepancy and Low-Dispersion Sequences," *Journal of Number Theory*, 30, 51-70.

13. Robinson, K.A. and Khuri, A.I. (2003), "Quantile Dispersion Graphs for Evaluating and Comparing Designs for Logistic Regression Models," *Computational Statistics and Data Analysis*, 43, 47-62.

14. Sitter, R.R. (1992), "Robust Designs for Binary Data," *Biometrics*, 48, 1145-1155.

15. Woods, D.C., Lewis, S.M., Eccleston, J.A. and Russell, K.G. (2005), "Designs for Generalized Linear Models with Several Variables and Model Uncertainty," *Technometrics*, in press.

16. Woods, D.C. (2005), "A Design Search Algorithm for Generalized Linear Models," http://www.maths.soton.ac.uk/staff/woods/glm_design.

APPENDIX: LOW-DISCREPANCY SEQUENCES

This appendix is intended to provide background on low-discrepancy sequences in general, and particulary on Nieddreiter's (1988) quasirandom sequence. Source code for an implementation (for MATLAB, C++ and Fortran90) can be found at http://www.csit.fsu.edu/~burkardt/m_src/niederreiter2/niederreiter2.html ; in addition to the source code the site briefly explains the nature of the algorithm:

"A quasirandom or low discrepancy sequence, such as the Faure, Halton, Hammersley, Niederreiter or Sobol' sequences, is 'less random' than a pseudorandom number sequence, but more useful for such tasks as approximation of integrals in higher dimensions, and in global optimization. This is because low discrepancy sequences tend to sample space 'more uniformly' than random numbers. Algorithms that use such sequences may have superior convergence."

We used NIEDERREITER2 which, as explained in the URL above, is an adapation of the INLO2 and GOLO2 routines in ACM TOMS Algorithm 738. The original code can only compute the "next" element of the sequence. The revised code allows the user to specify the index of any desired element. The original, true, correct version of ACM TOMS Algorithm 738 is available in the TOMS subdirectory of the NETLIB web site.

An Illustration

Figure A.1 compares 100 pseudorandom observations on $[0,1]^3$, produced by the command "RANDOM=rand(100,3)" in MATLAB (The MathWorks inc.), to a 3 dimensional Niederreiter base 2 low-discrepancy sequence with 2^{12} used as a seed, produced with the code suggested above. The upper row of the figure contains the 2-dimensional projections of the pseudorandom sequence, and the bottom row has the corresponding projections for Niederreiter's quasi-random sequence.

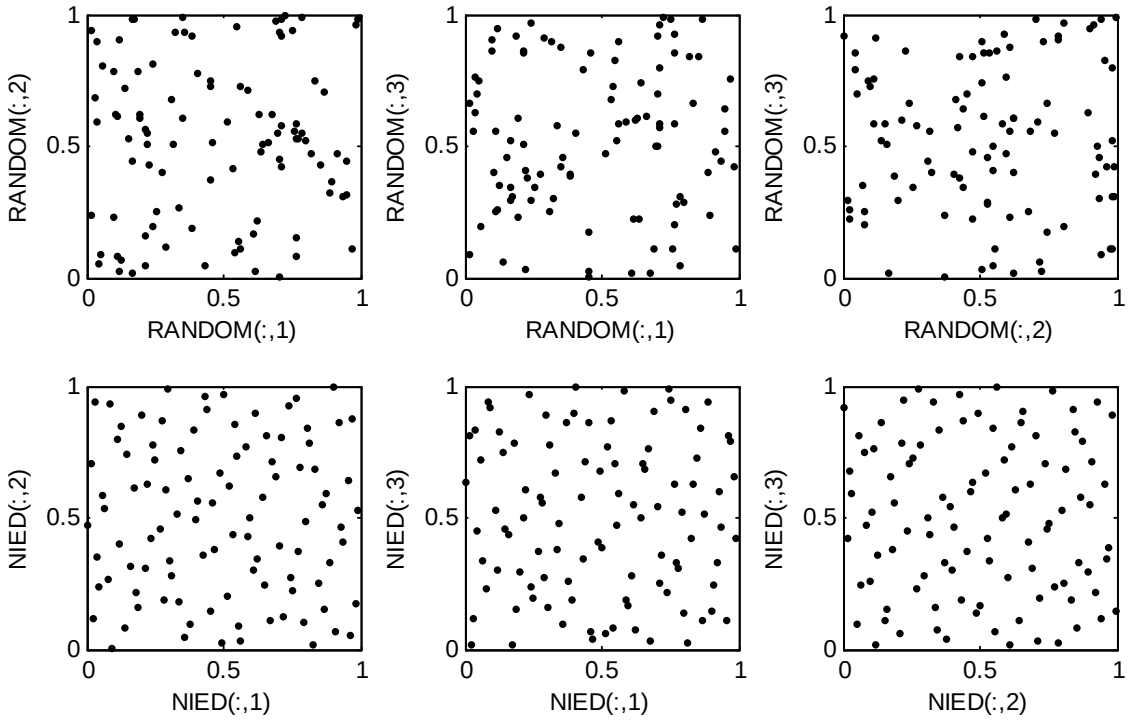

Figure A.1: Comparison of 2D projections of a pseudorandom sequence (top row) and a Niederreiter's quasi-random sequence (bottom row)

Clearly, the low-discrepancy sequence covers the space more evenly, avoiding empty gaps which are common in the pseudorandom sequence.

A brief overview on the mathmatical foundations of low-discrepency sequences can be found from Wikipedia, The Free Encyclopedia, at http://en.wikipedia.org/w/index.php?title=Low-discrepancy_sequence&oldid=27681750 ; the rest of this appendix is a part of the description in the quoted link.

A low-discrepancy sequence is a sequence with the property that for all N, the subsequence $x_1, ..., x_N$ is almost uniformly distributed (in a sense to be made precise), and $x_1, ..., x_{N+1}$ is almost uniformly distributed as well. Low-discrepancy sequences are also called quasi-random or sub-random sequences, due to their use in situations similar to those when pseudorandom or random numbers are used instead. The "quasi" modifier is used to denote more clearly that the numbers are not random (and to differentiate them from pseudorandomness, which uses different assumptions), but have useful properties similar to randomness in certain applications such as the quasi-Monte Carlo method.

The notion of uniformity is made precise as the discrepancy defined below. Roughly speaking, the discrepancy of a sequence is low if the number of points falling into a set B is close to the number one would expect from the measure of B. At least three methods of numerical integration can be phrased as follows. Given a set $x_1, ..., x_N$ in the interval $[0, 1]$, approximate the integral of a function f as the average of the function evaluated at those points: $\int_0^1 f(u)\, du \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i)$. If the points are chosen as $x_i = i/N$, this is the rectangle rule. If the points

are chosen to be randomly (or pseudorandomly) distributed, this is the Monte Carlo method. If the points are chosen as elements of a low-discrepancy sequence, this is the quasi-Monte Carlo method. A remarkable result, the Koksma-Hlawka inequality, shows that the error of such a method can be bounded by the product of two terms, one of which depends only on f, and another which is the discrepancy of the set $x_1, ..., x_N$.

It is convenient to construct the set $x_1, ..., x_N$ in such a way that if a set with $N+1$ elements is constructed, the previous N elements need not be recomputed. The rectangle rule uses points set which have low discrepancy, but in general the elements must be recomputed if N is increased. Elements need not be recomputed in the Monte Carlo method if N is increased, but the point sets do not have minimal discrepancy. By using low-discrepancy sequences, the quasi-Monte Carlo method has the desirable features of the other two methods.

Definition of discrepancy

The Star-Discrepancy is defined as follows, using Niederreiter's notation.

$D_N^*(P) = \sup_{B \in J^*} \left| \frac{A(B;P)}{N} - \lambda_s(B) \right|$ where P is the set $x_1, ..., x_N$, $\lambda_s$is the s-dimensional Lebesgue measure, $A(B;P)$ is the number of points in P that fall into B, and J^* is the set of intervals of the form $\prod_{i=1}^{s} [0, u_i)$ where u_i is in the half-open interval $[0, 1)$. Therefore $D_N^*(P) = \|\text{disc}\|_\infty$ where the discrepancy function is defined by $\text{disc}(y) = \frac{A([0,y);P)}{N} - \lambda_s([0, y))$

Two main conjectures

Conjecture 1. There is a constant c_s depending only on s, such that $D_N^*(x_1, ..., x_N) \geq c_s \frac{(\ln N)^{s-1}}{N}$ for any finite point set $x_1, ..., x_N$.

Conjecture 2. There is a constant c_s' depending only on s, such that $D_N^*(x_1, ..., x_N) \geq c_s' \frac{(\ln N)^s}{N}$ for any infinite sequence $x_1, x_2, x_3, ...$

These conjectures are equivalent. They have been proved for $s \leq 2$ by W. M. Schmidt. In higher dimensions, the corresponding problem is still open. The best-known lower bounds are due to K. F. Roth.

The best-known sequences

Constructions of sequences are known (due to Faure, Halton, Hammersley, Sobol', Niederreiter and Van der Corput) such that $D_N^*(x_1, ..., x_N) \leq C \frac{(\ln N)^s}{N}$ where C is a certain constant, depending on the sequence. After Conjecture 2, these sequences are believed to have the best possible order of convergence.