An Exchange Algorithm to find Designs under Generalized Linear Models

D.C. Woods*
Southampton Statistical Sciences Research Institute
University of Southampton

Introduction

The search algorithm presented here is capable of finding exact designs for generalized linear models, either locally D-optimal designs under a given model or robust designs which compromise across a class of models. An exchange algorithm is used to find designs from a discrete candidate list of possible design points. Background and details of the statistical methods are given in Woods, Lewis, Eccleston and Russell (2005), available at
`http://www.maths.soton.ac.uk/staff/woods/glm_design`.

Exchange algorithms

Exchange algorithms are the most common form of computer design search techniques. Although available in many flavours (see Nguyen and Miller (1992) for a review), all exchange algorithms share the same basic operations. Points in the current design are *exchanged* with those in a *candidate list* of possible design points. Exchanges which improve the performance of a design (as measured by an *objective function*) are accepted; otherwise the exchange is rejected. In common with other *greedy* algorithms, exchange algorithms do not allow the system to escape from local optima in the design space. Therefore, many runs, or tries, of the algorithm should be employed and the best design from across these tries selected.

The choice of a discrete candidate list of possible design points is an important factor in both the speed of the design search and the quality of the final design found by an exchange algorithm. Although this may sometimes result in sub-optimal solutions being found compared to an optimization technique that searches a continuous space of designs, such as simulated annealing (see, for example, Spall (2003), Ch. 8), it also has several advantages. Incorporation of constraints onto the design space is straightforward, as in mixture experiments when the variables are proportions that must sum to one (see Cornell, 2002) or when the allowable combinations of variable values are necessarily pre-determined. An example of this later situation is chemistry experiments where some variables are *descriptors* that give quantitative measurements of a qualitative factor (see, for example, Carlson, 1992). The allowable combinations of the descriptors are determined by the underlying levels of the qualitative factor. An exchange algorithm with a suitable candidate list is then an appropriate way to find efficient designs as demonstrated in, for example, McNamara et al (2005).

Modified Fedorov algorithm

Our algorithm is an implementation of a modified Fedorov exchange algorithm (Cook and Nachtsheim, 1980). The steps of the algorithm are as follows:

1. An initial starting design is chosen at random from the candidate list and its performance evaluated.

*Email: `D.C.Woods@maths.soton.ac.uk`
WWW: `http://www.maths.soton.ac.uk/staff/woods`

2. Each point in the current design is exchanged with each point in the candidate list in turn. For each exchange, the performance of the new design is assessed. Any exchanges which improve the design are accepted immediately; otherwise they are rejected.

3. The improvement in the objective function value after all exchanges have been considered is greater than a pre-specified cut-off δ, then the algorithm stops. Otherwise, step 2 is repeated.

Compiling and running

The algorithm is written in C++ and was developed under Linux using the g++ compiler. It requires the Gnu Scientific Library (GSL) (`http://www.gnu.org/software/gsl/`). A typical commandline to compile the code on a Linux system with the GSL installed in its default location would be

```
g++ -o exchange_search.run exchange_glm_search.cpp -lm -lgsl -lgslcblas
```

Input and output

By default, three input files are required. The first input file, input.in in glm_search_EA.zip, specifies the filenames for the main input file, the candidate list input file and the output file, as well as the random number seed (for repeatable searches). Example main input (exch_search.in) and candidate list files (cand.in) are provided in glm_search_EA.zip. Annotated versions of these files, exch_search.inREADME and cand.inREADME, are also provided and give detailed explanations of the input commands.

C++ classes

Nine different C++ classes are used in the algorithm code and are included in the zip file glm_search_SA.zip.

- matrix_class - a class of 2D arrays that interfaces with the GSL matrix routines
- matrixoperations_class - a class of static utilities for use with matrix_class
- random_class - a static utility class for the generation of pseudo-random numbers from various distributions using the GSL routines
- cont_design_class - objects of this class hold designs under a particular model
- model_class - these objects hold the link function and model parameters
- criteria_class - static utility class with methods for evaluating the objective functions for the D- and A- optimality criteria
- searchmisc_class.cpp - static utility class with a method to sort designs according to the value of the objective function
- exchange_search_class - static utility class with methods for exchange algorithm design search

Acknowledgements

This work took place as part of the Comb*e*chem e-Science project (EPSRC grant GR/R67729).

References

Carlson, R. (1992). *Design and Optimization in Organic Synthesis*. Elsevier, Amsterdam.

Cook, R.D. and Nachtsheim, C.J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22, 315-324.

Cornell, J.A. (2002). *Experiments with Mixtures: Designs, Models and the Analysis of Mixture Data*. Wiley, New York.

McNamara, C.A., Woods, D.C., Lewis, S.M., Bradley, M. and Frey J.G. (2005). Optimization of asymmetric dihydroxylation through optimal designed experiments. Technical Report 386, School of Mathematics, University of Southampton.

Nguyen, N. and Miller, A. (1992). A review of some exchange algorithms for constructing discrete D-optimal designs. *Computational Statistics and Data Analysis*, 14, 489-498.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Wiley, New York.

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.