The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$ 98	1-year subscription	$138
2-year subscription	$165	2-year subscription	$245
3-year subscription	$225	3-year subscription	$345
1-year student subscription	$ 75	1-year student subscription	$ 99
1-year institutional subscription	$245	1-year institutional subscription	$285
2-year institutional subscription	$445	2-year institutional subscription	$525
3-year institutional subscription	$645	3-year institutional subscription	$765
Electronic only		**Electronic only**	
1-year subscription	$ 75	1-year subscription	$ 75
2-year subscription	$125	2-year subscription	$125
3-year subscription	$165	3-year subscription	$165
1-year student subscription	$ 45	1-year student subscription	$ 45

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2014)
14, Number 3, pp. 605–622

Space-filling location selection

Michela Bia
CEPS/INSTEAD
Esch-sur-Alzette, Luxembourg
michela.bia@ceps.lu

Philippe Van Kerm
CEPS/INSTEAD
Esch-sur-Alzette, Luxembourg
philippe.vankerm@ceps.lu

Abstract. In this article, we describe an implementation of a space-filling location-selection algorithm. The objective is to select a subset from a list of locations so that the spatial coverage of the locations by the selected subset is optimized according to a geometric criterion. Such an algorithm designed for geographical site selection is useful for determining a grid of points that "covers" a data matrix as needed in various nonparametric estimation procedures.

Keywords: st0353, spacefill, spatial sampling, space-filling design, site selection, nonparametric regression, multivariate knot selection, point swapping

1 Introduction

Spatial statistics often address geographical sampling from a set of locations for networks construction (Cox, Cox, and Ensor 1997), for example, for installing air quality monitoring (Nychka and Saltzman 1998) or for evaluating exposure to environmental chemicals (Kim et al. 2010). The issue involves evaluating a discrete list of potential locations and determining a small, "optimal" subset of places—a "design"—at which to position, say, measurement instruments or sensors. One strategy to address such a problem—the geometric approach—aims to find a design that minimizes the aggregate distance between the locations and the sensors.

As discussed in Ruppert, Wand, and Carroll (2003) and Gelfand, Banerjee, and Finley (2012), location selection is also relevant in estimation of statistical models such as multivariate nonparametric or semiparametric regression models. By analogy, instead of locating measurement instruments, one seeks to identify a small number of "locations" from a large dataset at which to estimate a statistical model to reduce computational cost. For example, kernel density estimates or locally weighted regression models (Cleveland 1979; Fan and Gijbels 1996) are typically calculated on a grid of points spanning the data range rather than over the whole input data points (and interpolation is used where needed). The location of knots in spline regression models is somewhat related; a small number of knots are selected instead of knots being placed at many (or all) potential distinct data points. Determining such a grid is relatively easy in one-dimensional models—for example, it is customary to locate knots at selected percentiles of the data. Choosing an appropriate multidimensional grid while preserving computational tractability is more complicated because merely taking combinations of unidimensional grids quickly inflates the number of evaluation points. In this context, Ruppert, Wand, and Carroll (2003) recommend applying a geometric space-filling design to identify grid points or knot locations.

In this article, we describe an implementation of an algorithm for space-filling spatial-design construction. The algorithm developed in Royle and Nychka (1998) selects a set of "design points" from a discrete set of "candidate points" such that the coverage of the candidate points by the design points is optimized according to a geometric coverage criterion.[1] The algorithm involves iterative "point swapping" between the candidate points and the design points until no swapping can further improve the coverage of the candidate points by the design points. The coverage criteria is geometric, but it is not restricted to spatial, two-dimensional data. The procedure can be used in miscellaneous settings when optimal subsampling of multivariate data is needed. Constraints are easily imposed by excluding or including particular locations in the design. A nearest-neighbor approximation makes the algorithm fast even for large samples.

We describe Royle and Nychka's (1998) algorithm in section 2 and its implementation in Stata in section 3. We illustrate several uses of the `spacefill` command in section 4. We show how it can be applied for generating a multidimensional grid of fixed size that optimally "covers" a dataset.

2 Geometric coverage criterion and the point-swapping algorithm

2.1 Geometric coverage criterion

The space-filling design selection considered here is based on optimization with respect to the geometric coverage of a set of data points. We refer to data points as "locations", although they are not restricted to geographic locations identified by spatial coordinates—in principle, any unidimensional or multidimensional coordinates can be used to "locate" points (see examples in section 4).

Following Royle and Nychka's (1998) notation, we let C denote a set of N candidate locations (the "candidate set"). We let D_n be a subset of n locations selected from C. D_n is a "design" of size n, and the locations selected in D_n are "design points". The geometric metric for the distance between any given location $\mathbf{x}$ and the design D_n is

$$d_p(\mathbf{x}, D_n) = \left(\sum_{\mathbf{y} \in D_n} ||\mathbf{x} - \mathbf{y}||^p \right)^{\frac{1}{p}} \tag{1}$$

with $p < 0$. $d_p(\mathbf{x}, D_n)$ measures how well the design D_n "covers" the location $\mathbf{x}$. When $p \longrightarrow -\infty$, $d_p(\mathbf{x}, D_n)$ tends to the shortest Euclidean distance between $\mathbf{x}$ and a point in D_n (Johnson, Moore, and Ylvisaker 1990). $d_p(\mathbf{x}, D_n)$ is zero if $\mathbf{x}$ is at a location in D_n.

1. An R implementation of Royle and Nychka's (1998) algorithm is available in Furrer, Nychka, and Sain (2013).

A design D_n^* is considered to optimally cover the set of locations C for parameters p and q if it minimizes

$$C_{p,q}(C, D_n) = \left\{ \sum_{\mathbf{x} \in C} d_p(\mathbf{x}, D_n)^q \right\}^{\frac{1}{q}} \tag{2}$$

over all possible designs D_n from C. The optimal design minimizes the q power mean of the "coverages" of all locations outside of the design (the candidate points). Increasing q gives greater importance to the distance of the design to poorly covered locations.

Figure 1 can help readers visualize the criterion. From a set of 38 European cities, we selected a potential design of five locations: Madrid, Brussels, Berlin, Riga, and Sofia. The coverage of, say, London by this design is given by plugging the Euclidean distances from London to the five selected cities into (1). With a large negative p, this coverage will be determined by the distance to the closest city, namely, Brussels. Repeating such calculations for all 33 cities from outside the design and aggregating the coverages using (2) gives the overall geometric "distances" of European cities to the design composed of Madrid, Brussels, Berlin, Riga, and Sofia. The optimal design is the combination of any five cities that minimizes this criterion. The design composed of Madrid, Brussels, Berlin, Riga, and Sofia is in fact the optimal design for $p = -5$ and $q = 1$.

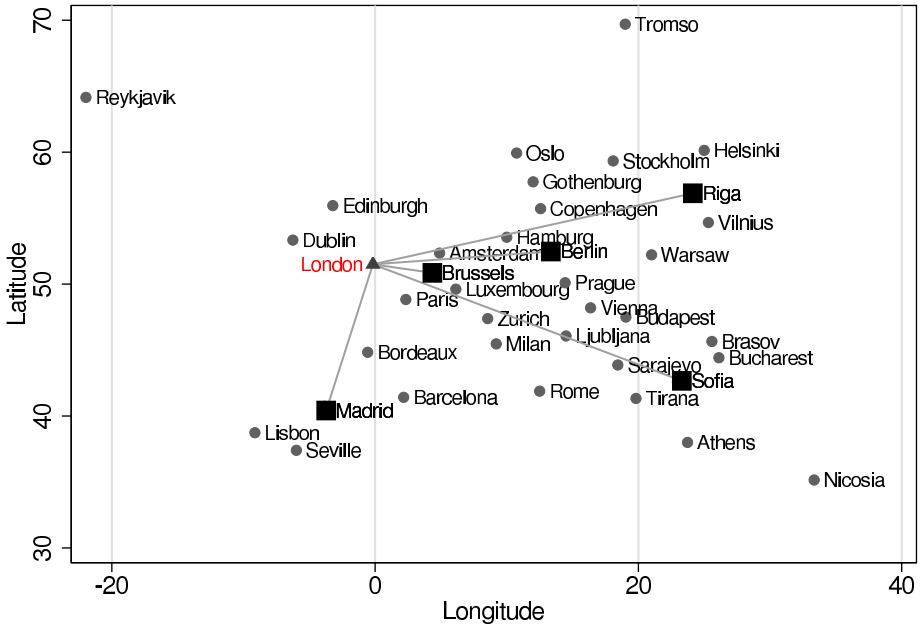

Figure 1. Coverage criterion illustration: Coverage of a five-city design (Madrid, Brussels, Berlin, Riga, and Sofia) with distances to London as example

2.2 A point-swapping algorithm

In most applications, identification of the optimal design by calculating the coverage criterion for all possible subsets of size n from N is computationally prohibitive. Royle and Nychka (1998) propose a simple point-swapping algorithm to determine D_n^*. Starting from a random initial design D_n^0, the algorithm iteratively attempts to swap a point from the design with the point from the candidate set that leads to the greatest improvement in coverage. If this tentative swap improves coverage of the candidate set by the design, the latter is updated. Otherwise, the swap is ignored. The process is repeated until no swap between a design point and a candidate point can improve coverage. Users can significantly improve speed by restricting potential swaps for a point in the design to its k nearest neighbors in the candidate set [according to (1)]. See Royle and Nychka (1998) for details.

The point-swapping algorithm makes it straightforward to impose constraints on the inclusion or exclusion of specific locations; such points are considered in calculations of the geometric criterion but excluded from any potential swap. Nonrandom initial design points can also be used.

Although the algorithm always converges to a solution, it is not guaranteed to converge to the globally optimal D_n^* for any initial design when potential swaps are limited to nearest neighbors. Therefore, Royle and Nychka (1998) recommend repeating estimation for multiple initial design sets and selecting the design with the best coverage across repetitions (see section 4).

3 The spacefill command

The `spacefill` command performs space-filling location selection using Royle and Nychka's (1998) point-swapping algorithm. It operates on N observations from variables identifying the coordinates of the data points and returns the subset of $n < N$ observations that optimally covers the data.

`spacefill` options allow forced inclusion or exclusion of particular observations, user-specified initial design, and automatic standardization of location coordinates. When weights are specified, `spacefill` performs weighted calculation of the aggregate coverage measure [see (2)]. In section 4, we show that combining weights and restrictions on candidate locations makes it easy to create an "optimal" regular grid over a dataset.

3.1 Syntax

`spacefill` *varlist* [*if*] [*in*] [*weight*] [`,` `ndesign(`#`)` `design0(`*varlist*`)`
`fixed(`*varname*`)` `exclude(`*varname*`)` `p(`#`)` `q(`#`)` `nnfrac(`#`)` `nnpoints(`#`)`
`nruns(`#`)` `standardize` `standardize2` `standardize3` `sphericize` `ranks`
`generate(`*newvar*`)` `genmarker(`*newvar*`)` `noverbose`]

`aweight`s, `fweight`s, and `iweight`s are allowed; see [U] **11.1.6 weight**.

varlist and the `if` or `in` qualifier identify the data from which the optimal subset is selected.

3.2 Options

`ndesign(`#`)` specifies n, the size of the design. The default is `ndesign(4)`.

`design0(`*varlist*`)` identifies a set of initial designs identified by observations with nonzero *varlist*. If multiple variables are passed, one optimization is performed for each initial design, and the selected design is the one with best coverage.

`fixed(`*varname*`)` identifies observations that are included in all designs when *varname* is nonzero.

`exclude(`*varname*`)` identifies observations excluded from all designs when *varname* is nonzero.

`p(`#`)` specifies a scalar value for the distance parameter for calculating the distance of each location to the design; for example, $p = -1$ gives harmonic mean distance, and $p = -\infty$ gives the minimum distance. The default is `p(-5)`, as recommended in Royle and Nychka (1998).

`q(`#`)` specifies a scalar value for the parameter q. The default is `q(1)` (the arithmetic mean).

`nnfrac(`#`)` specifies the fraction of data to consider as nearest neighbors in the point-swapping iterations. Limiting checks to nearest neighbors improves speed but does not guarantee convergence to the best design; therefore, setting `nruns(`#`)` is recommended. The default is `nnfrac(0.50)`.

`nnpoints(`#`)` specifies the number of nearest neighbors considered in the point-swapping iterations. Limiting checks to nearest neighbors improves speed. `nnfrac(`#`)` and `nnpoints(`#`)` are mutually exclusive.

`nruns(`#`)` sets the number of independent runs performed on alternative random initial designs. The selected design is the one with best coverage across the runs. The default is `nruns(5)`.

`standardize` standardizes all variables in *varlist* to zero mean and unit standard deviation (SD) before calculating distances between observations.

`standardize2` standardizes all variables in *varlist* to zero mean and SD before calculating distances between observations, with an estimator of the SD as 0.7413 times the interquartile range.

`standardize3` standardizes all variables in *varlist* to zero median and SD before calculating distances between observations, with an estimator of the SD as 0.7413 times the interquartile range.

`sphericize` transforms all variables in *varlist* into zero mean, SD, and zero covariance using a Cholesky decomposition of the variance–covariance matrix before calculating distances between observations.

`ranks` transforms all variables in *varlist* into their (fractional) ranks and uses distances between these observation ranks in each dimension to evaluate distances between observations.

`generate(`*newvar*`)` specifies the names for new variables containing the locations of the best design points. If one variable is specified, it is used as a *stubname*; otherwise, the number of new variable names must match the number of variables in *varlist*.

`genmarker(`*newvar*`)` specifies the name of a new binary variable equal to one for observations selected in the best design and zero otherwise.

`noverbose` suppresses output display.

Options `standardize2`, `standardize3`, and `ranks` require installation of the user-written package `moremata`, which is available on the Statistical Software Components archive (Jann 2005).

4 Examples

We provide two illustrations for the application of `spacefill`. The first example uses `ozone2.txt`, which is available in the R *fields* package (Furrer, Nychka, and Sain 2013), and provides examples of standard site selection. The second example uses survey data from the *Panel Socio-Économique Liewen zu Lëtzebuerg*/European Union-Statistics on Income and Living Conditions (PSELL3/EU-SILC) and illustrates the use of `spacefill` for nonparametric regression analysis with multidimensional, nonspatial data.

4.1 Basic usage

`ozone2.txt` contains air quality information in 147 locations in the US Midwest in the Summer 1987 (Furrer, Nychka, and Sain 2013). Locations are identified by their relative latitude (`lat`) and longitude (`lon`).

We start by selecting an optimal design of size 10 from the 147 locations, using default values $p = -5$ and $q = 1$, candidate swaps limited to the nearest half of the locations, and 5 runs with random starting designs.

```
. insheet using ozone2.txt
(3 vars, 147 obs)

. spacefill lon lat, ndesign(10)
Run 1  ....                                    (Cpq =         100.34)
Run 2  ....                                    (Cpq =          96.92)
Run 3  ......                                  (Cpq =          94.19)
Run 4  ....                                    (Cpq =          95.00)
Run 5  ..                                      (Cpq =          95.19)

. return list

scalars:
                  r(q) =  1
                  r(p) =  -5
                 r(nn) =  69
                r(Cpq) =  94.19164847896585
          r(nexcluded) =  0
             r(nfixed) =  0
           r(ndesign) =  10
                 r(N) =  147

macros:
            r(varlist) : "lon lat"

matrices:
        r(Best_Design) :  10 x 2

. matrix list r(Best_Design)

r(Best_Design)[10,2]
            lon         lat
 r1  -87.752998      41.855
 r2  -90.160004      38.612
 r3  -85.841003   39.935001
 r4      -87.57      38.021
 r5  -91.662003   41.992001
 r6  -84.476997   39.106998
 r7  -85.578003   38.137001
 r8  -85.671997   42.985001
 r9     -83.403      42.388
r10  -88.283997      43.333
```

Notice that the first run leads to a somewhat higher aggregate distance to the design points (`Cpq=100.34`) than the other runs. This stresses the importance of multiple starting designs. Figure 1 shows the selected locations in the best design (achieved at run 3, where `Cpq=94.19`).

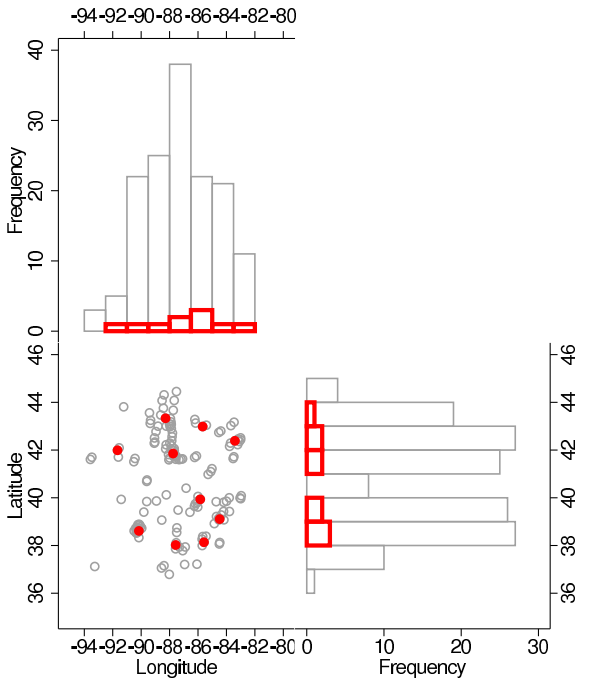

Figure 2. Scatterplot and histogram of longitude and latitude for all 147 locations (gray histograms and gray hollow circles) and 10 best design points (thick histograms and solid dots) with $p = -5$ and $q = 1$ (default)

Users can improve speed by restricting potential swaps to a smaller number of nearest neighbors. Limiting a search to 25 nearest neighbors (against 69—the default half of the locations—in the first example), our second example below runs in 4 seconds against 11 seconds for our initial example, without much loss in the coverage of the resulting design (`Cpq=96.59`). On the other hand, running `spacefill` with the full candidates as potential swaps runs in over 30 seconds for an optimal design with `Cpq=91.96`.

```
. spacefill lon lat, ndesign(10) nnpoints(25) genmarker(set1)
Run 1  .....                                (Cpq =         117.02)
Run 2  ....                                 (Cpq =         109.93)
Run 3  ..                                   (Cpq =         110.99)
Run 4  ..                                   (Cpq =         101.05)
Run 5  .....                                (Cpq =          96.59)
. spacefill lon lat, ndesign(10) nnfrac(1)
Run 1  ...                                  (Cpq =          91.96)
Run 2  ....                                 (Cpq =          91.96)
Run 3  ..                                   (Cpq =          91.96)
Run 4  ...                                  (Cpq =          92.32)
Run 5  ...                                  (Cpq =          91.96)
```

We now illustrate the use of the `genmarker()`, `fixed()`, and `exclude()` options. In the previous call, `genmarker(set1)` generated a dummy variable equal to 1 for the 10

points selected into the best design and 0 otherwise. We now specify `exclude(set1)` to derive a new design with 10 different locations and then use `fixed(set2)` to force this new design into a design of size 15.

```
. spacefill lon lat, ndesign(10) nnpoints(25) exclude(set1) genmarker(set2)
> noverbose
10 points excluded from designs (set1>0)
. spacefill lon lat, ndesign(15) nnpoints(25) fixed(set2) genmarker(set3)
> noverbose
10 fixed design points (set2>0)
. list set1 set2 set3 if set1+set2+set3>0
```

	set1	set2	set3
4.	1	0	0
10.	0	1	1
25.	1	0	0
40.	1	0	0
48.	0	1	1
55.	1	0	0
58.	0	1	1
60.	1	0	0
61.	0	1	1
63.	0	0	1
67.	0	0	1
74.	1	0	0
77.	0	0	1
80.	0	1	1
82.	0	1	1
89.	0	0	1
91.	0	1	1
97.	1	0	0
107.	0	1	1
109.	1	0	0
121.	0	1	1
125.	0	0	1
135.	0	1	1
140.	1	0	0
143.	1	0	0

The key parameters q and p of the coverage criterion can also be flexibly specified. Figure 2 illustrates 3 designs selected with default parameters $p = -5$ and $q = 1$ (dots), with $p = -1$ and $q = 1$ (squares), and with $p = -1$ and $q = 5$ (crosses). With $p = -5$, the distance of a location to the design is mainly determined by the distance to the closest point of the design; $p = -1$ accounts for the distance to all points in the design, leading to more central location selections. Setting $q = 5$ penalizes large distances between design and nondesign points, leading to location selections more spread out toward external points. Note our use of user-specified random starting designs with option `design0()` to ensure comparison is made on common initial values.

```
. generate byte init1 = 1 in 1/10
(137 missing values generated)
. generate byte init2 = 1 in 11/20
(137 missing values generated)
. generate byte init3 = 1 in 21/30
(137 missing values generated)
. generate byte init4 = 1 in 31/40
(137 missing values generated)
. generate byte init5 = 1 in 41/50
(137 missing values generated)
. local options nnfrac(0.3) nruns(10) design0(init1 init2 init3 init4 init5)
> noverbose
. spacefill lat lon, `options´ generate(Des)
. spacefill lat lon, `options´ generate(Des_BIS) p(-1) q(1)
. spacefill lat lon, `options´ generate(Des_TER) p(-1) q(5)
. spacefill lat lon, `options´ generate(Des_QUAT) p(-5) q(5)
```

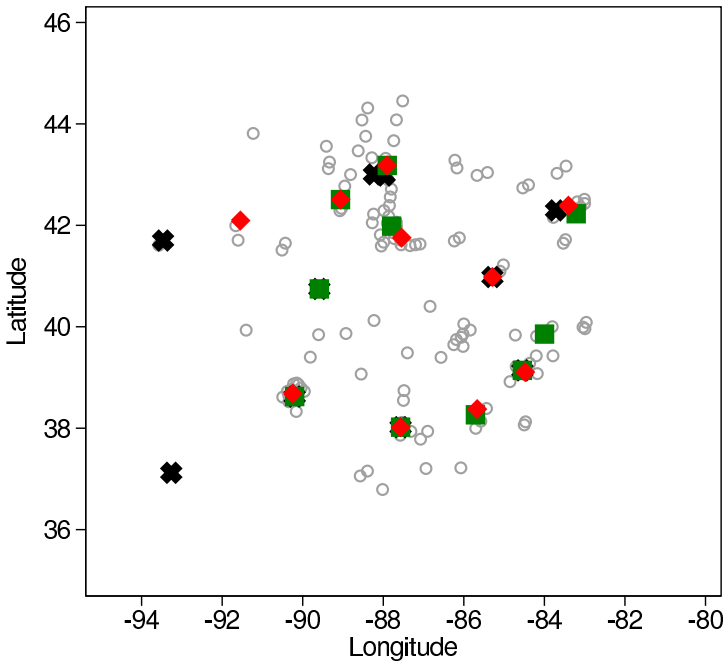

Figure 3. Scatterplot of longitude and latitude for all 147 locations (gray hollow circles) and best design points with default $p = -5$ and $q = 1$ (dots), with $p = -1$ and $q = 1$ (squares), and with $p = -1$ and $q = 5$ (crosses)

4.2 Design selection from external locations: Lattice subsets

By combining the `exclude()` option and weights, one can use `spacefill` to find an optimal design from an external set of locations; that is, one can use it to select a subset of points from a set A that optimally covers points from a set B. This is particularly useful to identify a subset of points from a lattice (the set A) that best covers the data (the set B). To set this up, we start by generating the lattice—a dataset with many candidate grid points—using `range` (see [D] **range**) and `fillin` (see [D] **fillin**). We append this generated dataset to the locations data. We then identify actual observations from the sample by `sample==0` and the generated candidate locations on the lattice by `sample==1`.

We can now run `spacefill` to select a smaller subset of grid points from the full lattice that optimally covers the actual locations. To do so, we run `spacefill` on the whole set of data points with i) `exclude(sample)` to select points from the grid only and ii) with `[iw=sample]` so that the aggregate distance is computed only between the design points on the grid and the actual locations. A set of 25 optimally chosen grid points from a candidate grid of 176 (11×16) points is shown in figure 3. Below we illustrate how this can be used to speed up calculations of computationally intensive nonparametric regression models.

```
. clear
. set obs 16
obs was 0, now 16
. range lon -95 -80 16
. range lat 36 46  11
(5 missing values generated)
. fillin lon lat
. gen byte sample = 0
. save gridlatlon.dta , replace
file gridlatlon.dta saved
. clear
. insheet using ozone2.txt
(3 vars, 147 obs)
. keep lat lon
. gen byte sample = 1
. append using gridlatlon
. spacefill lon lat [iw=sample], exclude(sample) ndesign(25) nnpoints(100)
> genmarker(subgrid1)
147 points excluded from designs (sample>0)
Run 1  ..                                    (Cpq =          63.93)
Run 2  ....                                  (Cpq =          63.92)
Run 3  ....                                  (Cpq =          63.71)
Run 4  ...                                   (Cpq =          63.07)
Run 5  ...                                   (Cpq =          63.02)
```

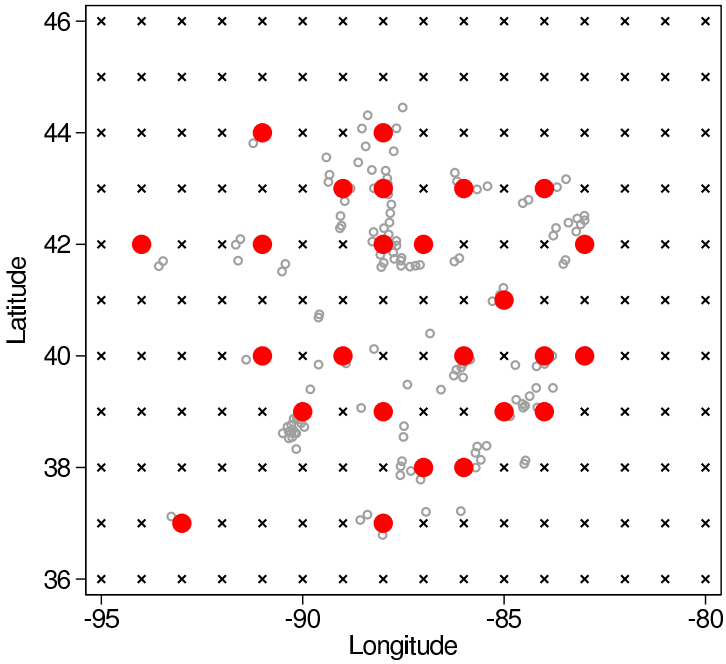

Figure 4. Actual 147 locations (hollowed gray circles), 176 candidate grid points (lattice; crosses), and 25 optimally selected grid points (solid dots)

4.3 Handling nonspatial data: Nonparametric regression example

We now illustrate the use of `spacefill` with multidimensional and nonspatial data taken from the PSELL3/EU-SILC collected in 2007.[2] We extracted information on the height, weight, and wage of a random subsample of 500 working women.

We first use `spacefill` to select a subset of 50 women with characteristics on these 3 variables that best "cover" the sample. Given the different metric of the three variables, we specify the `standardize` option to compute the geometric distance criterion after standardizing the three variables to have zero mean and unit SD in the sample.[3]

Figures 5 and 6 show bivariate scatterplots and histograms of the selected 50 design points. Two features are worth noting. First, the quality of the coverage is not affected by the skewness of the data (especially in the wage dimension). The space-filling algo-

2. PSELL3/EU-SILC is a longitudinal survey on income and living conditions representative of the population residing in Luxembourg. Data are collected annually in a sample of more than 3,500 private households.
3. Alternative standardization could have been adopted with options `standardize2`, `standardize3`, `sphericize`, or `ranks`.

rithm is indeed applicable to broad data configurations. Second, the difference in the histograms for the sample and for the design points is a reminder that selecting a space-filling design is distinct from drawing a "representative subset" of the data. The points that best cover the data in a geometric sense must not necessarily reflect their frequency distribution: few design points may contribute to cover many data points in areas of high concentration, while design points spread out in areas of low data concentration will contribute to cover a smaller number of data points.

```
. summarize height weight wage
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      height |        500      165.21      6.8886       150        192
      weight |        500      65.368    12.80502        43        127
        wage |        500    2720.688    1920.047       300      10000

. spacefill height weight wage, ndesign(50) nnfrac(0.05) generate(BH BW BWa)
> standardize
Run 1  ....                                  (Cpq =         196.98)
Run 2  .....                                 (Cpq =         195.15)
Run 3  ....                                  (Cpq =         196.13)
Run 4  ........                              (Cpq =         196.79)
Run 5  ....                                  (Cpq =         194.55)
```

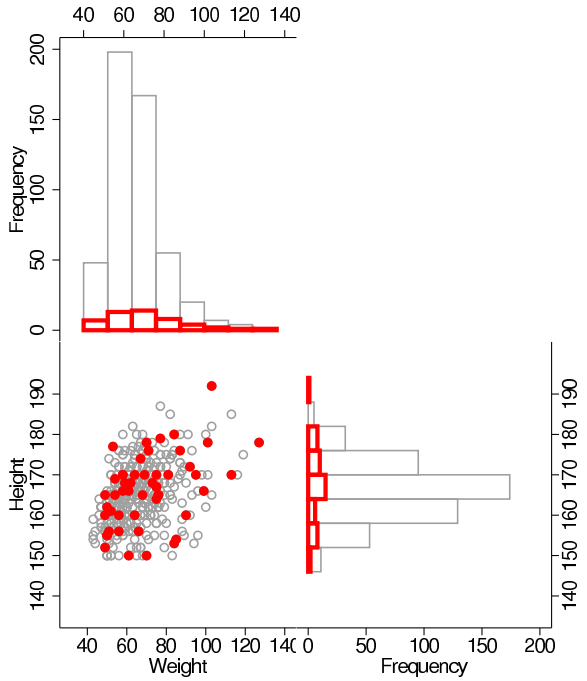

Figure 5. Scatterplot and histogram of height and weight for all data (gray histograms and hollowed markers) and best design points (thick histograms and markers) for the standardized values of the height, weight, and wage

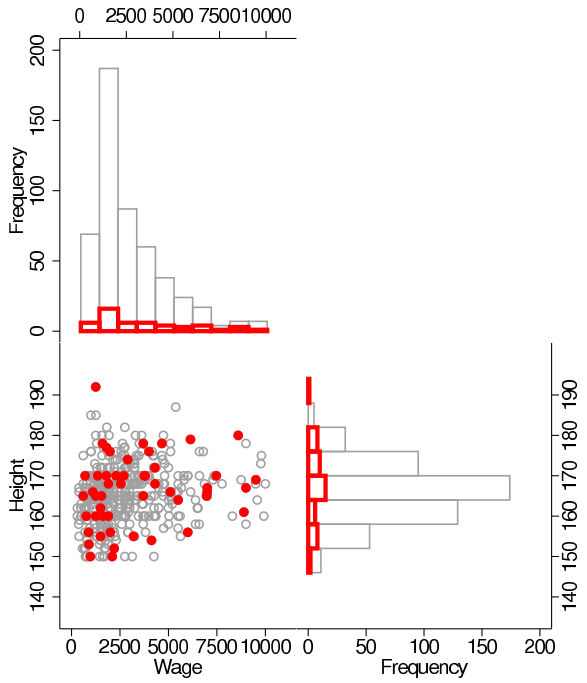

Figure 6. Scatterplot and histogram of height and wage for all data (gray histograms and hollowed markers) and best design points (thick histograms and markers) for the standardized values of the height, weight, and wage

We now use these data to run a locally weighted polynomial regression of wage on height and weight. Our objective is to assess nonparametrically the relationship between wage and body size. For the sake of illustration, we want to estimate expected wage nonparametrically at multiple grid points from a lattice where each point is a pair of height–weight values. One reason for this is that fitting the model at all height–weight pairs in our data would be computationally expensive (and inefficient if there are nearly identical height–weight pairs in the data). We seek a cheaper alternative with fewer evaluation points. (This is similar to using `lpoly` with the `at()` option instead of `lowess` in the unidimensional setting.) Also we use evaluation points on a lattice instead of "at sample values" because we are considering fitting the model for different subsamples, and we want to have model estimates on a common grid of evaluation points for all subsamples. (If need be, bivariate interpolation will be used to recover estimates at sample values; see [G-2] **graph twoway contourline** for the interpolation formula.) This setting is relatively standard in nonparametric regression analysis, especially when dealing with large samples or computationally heavy estimators (for example, cross-validation-based bandwidth selection).

We start with a 20×20 rectangular lattice covering heights from 150 to 192 centimeters and weights from 43 to 127 kilograms. While this lattice spans the values observed in our sample, it also includes many empirically irrelevant height–weight pairs. Estimation on the full grid is therefore unnecessary, and we use `spacefill` as described above to select a subset of points on the lattice that covers our data.

Figure 7 shows resulting estimates based on a space-filling design of size 50, as well as estimates based on a random subset of 100 lattice points, on 100 Halton draws from the lattice, on the full lattice, and on all sample points. Brightness of the contours corresponds to local regression estimates of expected wage from black (for monthly wage below EUR 1000) to white (for monthly wage above EUR 5000). In each panel, local regression was effectively calculated only at the marked grid points (and so it was conducted faster on the space-filling design), while the overall coloring of the map was based on the thin-plate-spline interpolation built in `twoway contour`.

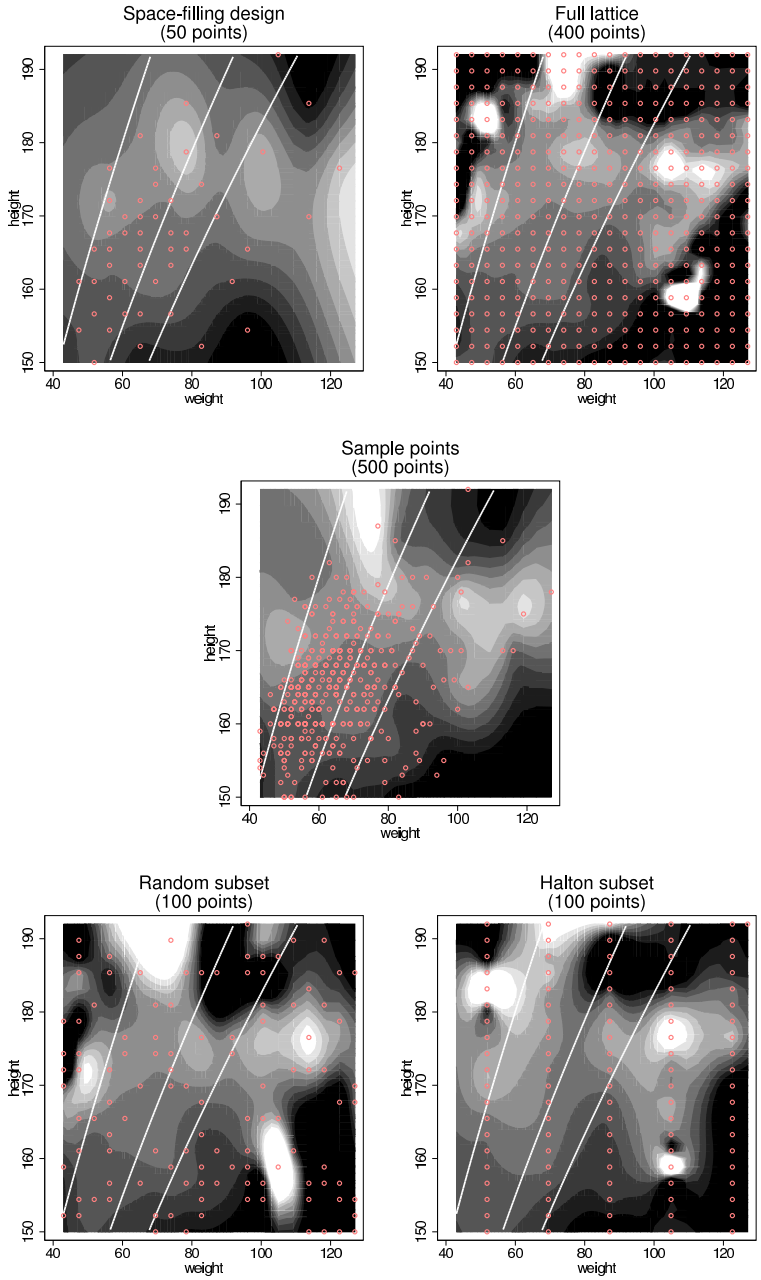

Figure 7. Contour plot of expected wage of 500 Luxembourg women by height and weight from monthly wage less than EUR 1000 (black) to more than EUR 5000 (white). Calculations based on local regression estimation. White lines identify body-mass indices of 18.5, 25, and 30, which delineate underweight, overweight, and obesity, respectively.

The contour plots display variations in areas of low data density (top left and bottom right), reflecting both the imprecision and variability of the local linear regression estimates in these zones and the variations introduced by the interpolation of values away from the bulk of the data. In areas of higher data density—for height below 180 centimeters and weight below 100 kilograms—estimates on the 50-points space-filling subset differ little from those of the full sample or from the full lattice.[4]

Acknowledgments

This research is part of the project "Estimation of direct and indirect causal effects using semi-parametric and non-parametric methods", which is supported by the Luxembourg "Fonds National de la Recherche", cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND). Philippe Van Kerm acknowledges funding for the project "Information and Wage Inequality", which is supported by the Luxembourg "Fonds National de la Recherche" (contract C10/LM/785657).

5 References

Cleveland, W. S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* 74: 829–836.

Cox, D. D., L. H. Cox, and K. B. Ensor. 1997. Spatial sampling and the environment: Some issues and directions. *Environmental and Ecological Statistics* 4: 219–233.

Fan, J., and I. Gijbels. 1996. *Local Polynomial Modelling and Its Applications*. New York: Chapman & Hall/CRC.

Furrer, R., D. Nychka, and S. Sain. 2013. fields: Tools for spatial data. R package version 6.7.6. http://CRAN.R-project.org/package=fields.

Gelfand, A. E., S. Banerjee, and A. O. Finley. 2012. Spatial design for knot selection in knot-based dimension reduction models. In *Spatio-Temporal Design: Advances in Efficient Data Acquisition*, ed. J. Mateu and W. G. Müller, 142–169. Chichester, UK: Wiley.

Jann, B. 2005. moremata: Stata module (Mata) to provide various functions. Statistical Software Components S455001, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s455001.html.

Johnson, M. E., L. M. Moore, and D. Ylvisaker. 1990. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* 26: 131–148.

Kim, J.-I., A. B. Lawson, S. McDermott, and C. M. Aelion. 2010. Bayesian spatial modeling of disease risk in relation to multivariate environmental risk fields. *Statistics in Medicine* 29: 142–157.

4. Note, incidentally, how taller women tend to be paid higher wages in these data in all three body-mass index categories.

Nychka, D., and N. Saltzman. 1998. Design of air-quality monitoring networks. In *Case Studies in Environmental Statistics (Lecture Notes in Statistics 132)*, ed. D. Nychka, W. Piegorsch, and L. Cox, 51–76. New York: Springer.

Royle, J. A., and D. Nychka. 1998. An algorithm for the construction of spatial coverage designs with implementation in SPLUS. *Computers and Geosciences* 24: 479–488.

Ruppert, D., M. P. Wand, and R. J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

About the authors

Michela Bia and Philippe Van Kerm are at CEPS/INSTEAD, Esch-sur-Alzette, Luxembourg.