The Stata Journal

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press, and Stata is a registered trademark of StataCorp LP.

The Stata Journal (2005)
5, Number 2, pp. 224–238

Stata in space: Econometric analysis of spatially explicit raster data

Daniel Müller
Institute for Agricultural Economics and Social Sciences
Humboldt University Berlin, Germany
d.mueller@agrar.hu-berlin.de

Abstract. Realizing the importance of location, economists are increasingly adopting spatial analytical and spatial econometric perspectives to study questions such as the geographical targeting of policy interventions, regional agglomeration effects, the diffusion of technologies across space, or causes and consequences of land-cover change. Explicitly accounting for location in econometric estimations can be of great benefit for researchers working at the interface of economics or environmental sciences and geography.

The objective of this article is to demonstrate how spatially explicit raster data derived from standard geographical information system (GIS) software can be used within Stata. Three programs implemented as ado-files are presented. These import geographic raster data into Stata (`ras2dta`), draw systematic spatial samples within Stata (`spatsam`), and export data and estimation results in a form usable by standard GIS software (`dta2ras`). A numerical example is presented to estimate the determinants of forest cover with a spatially explicit logit model, calculate predicted probabilities, and map the predictions with GIS software.

Keywords: dm0014, ras2dta, spatsam, dta2ras, geographical information systems (GIS), raster data, spatial modeling, spatial econometrics

1 Background

Socioeconomic data typically exist for discrete social entities, such as administrative units or households, and are often not linked to the geographical location. Conversely, data stemming from disciplines of natural sciences are often collected including geographical referencing, or georeferencing, which involves the assignment of coordinates defined by a spatial reference or coordinate system to objects like points, lines, or polygons. A common example for georeferenced data is the use of meteorological stations that measure climatic indicators like rainfall used to derive spatially continuous rainfall surfaces from spatial interpolations of point measurements. Other examples of geographic data relevant to a wide variety of studies in environmental sciences are temperature, slope, elevation, and soil qualities. The integration of georeferenced socioeconomic and natural science data can facilitate interdisciplinary approaches and provide additional insights into numerous statistical applications that are influenced by underlying spatial processes or spatial relationships.

Spatial econometrics deal with the analysis of economic data that is explicitly linked to location. The techniques of spatial econometrics account for the peculiarities introduced by the spatial perspective and are justified based on two reasons: First, *spatial heterogeneity* might arise due to a lack of structural stability across space, such as varying parameters or functional forms, and due to nonhomogeneity of the units of observations across space (Anselin 1988). Second, *spatial autocorrelation*—methodologically similar to autocorrelation in time-series models—refers to a lack of independence among observations. Spatial autocorrelation pertains to a coincidence of value similarity with locational similarity (Anselin 1988). This dependence among observations and the importance of relative locations is expressed by Tobler (1979) in his first law of geography, which states that "everything is related to everything else, but near things are more related than distant things". Interactions among neighboring agents could, for example, induce a correlation of the variables across space, which must be accounted for in model estimation.

The various spatial relationships among observations can result in unreliable estimates and incorrect statistical inference of the parameters (Anselin 1988). For many social and economic processes, a better appreciation of the spatial context can potentially avoid misleading inferences and improve the strength of results and their interpretation. Knowledge about the location of a process and its interaction with processes at neighboring locations can help infer the underlying reasons and logic of the process under investigation. However, spatial analysis adds mathematical complexity due to the necessary incorporation of two dimensions (in X- and in Y-direction).

The next section briefly introduces geographic information systems (GIS) and the structure of raster data. Section 3 outlines the program to import raster grids into Stata. Section 4 presents the program for systematic spatial sampling from a raster surface, and section 5 the program to export the Stata files into a format usable by standard GIS software. The last section presents a numerical example in which I estimate the determinants to observe forest cover using a spatially explicit binary logit model.

2 Geographic information systems and the raster data model

A geographic information system serves to compile, store, manipulate, analyze, and visualize spatial data. As the two main data models in a GIS, the *vector* and the *raster* model are distinguished. Vector data contain X- and Y-coordinates, which represent points (single X- and Y-coordinates), lines (series of ordered points), and polygons (closed lines with equal start and end coordinate). The raster data model is represented as an arrangement of regularly shaped, contiguous cells in a two-dimensional matrix, which together form a continuous data layer. A layer typically consists of square cells, which fit together edge-to-edge. Each cell represents one location in a raster surface and contains integer or floating-point numbers indicating the characteristic of that location. A dataset usually contains various layers (or bands), which are stacked on top of each other and cover the geographical area of interest. A common application of multiple-

layer data are multispectral satellite images where various bands cover a certain spectral range of reflected electromagnetic energy. Raster data has the advantage of conceptual simplicity, compact data storage, and well-established algorithms for processing and analyzing. A main disadvantage is that it artificially imposes grid cell borders on continuous phenomena, which is often better represented in the vector model.

The structure of a raster data model is sketched in figure 1. Figure 1 (a) shows data in a 5×5 matrix of square cells with discrete observations ranging from one to five. The corresponding map in figure 1 (b) on the right side can be exported from common GIS software packages in ASCII data format.

(a) **(b)**

1	1	2	2	2
3	1	3	2	1
2	3	3	4	2
3	5	5	1	3
2	2	4	4	5

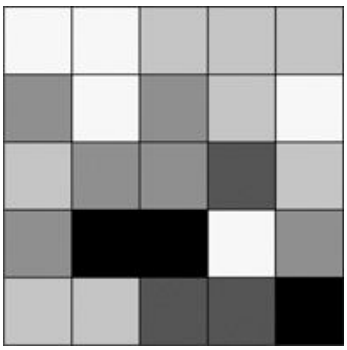

Figure 1: Raster data structure (a) and corresponding raster map (b)

The exported ASCII file contains the spatial information in a header that occupies the first six rows of the data (see table 1). The number of columns (ncols = 5) is indicated in the first line and the number of rows (nrows = 5) in the second line. Line three and four locate the map in space with geographic coordinates for the lower left X-coordinate, xllcorner, and for the lower left Y-coordinate, yllcorner. Line five states the size of the cells in the specified map units, and line six assigns numerical missing values (ArcInfo™ and ArcView™ assign -9999 by default). The map values start in line six with the upper-left cell from figure 1. Values are separated by spaces and move from left to right, and then top to bottom. No carriage returns are necessary, as the number of columns in the header information determines when a new row starts (Environmental Systems Research Institute (ESRI) 2000).

(*Continued on next page*)

Table 1: ASCII raster with header information

ncols	5
nrows	5
xllcorner	1
yllcorner	1
cellsize	1
NODATA_value	−9999
1 1 2 2 2 3 1 3 2 1 2 3 3 4 ...	

3 Importing data

The structure of the ASCII file from figure 1 and table 1 makes it straightforward to import the text file into Stata. The program `ras2dta` imports the data starting from observation 13 after the header that ends with the default missing value of −9999. Each raster cell of a map is turned into one observation, and an entire map of raster cells yields one variable in Stata. Figure 1 yields 25 observations starting from the top left, to the left, then to the bottom (first observation = 1, second = 1, ..., sixth = 3, seventh = 1, ..., twenty-fifth = 5).

Information from the header is used to identify the missing values and the number of X- and Y-coordinates. `ras2dta` optionally generates a variable carrying an identity code for the cells, `idcell()`, to later facilitate the use of, e.g., `merge` or `joinby` and the export of data back to the GIS software. Missing values, like values located outside the area of interest, can be dropped when the data is read by `infile`. Optionally, two variables are generated representing the X- and Y-indicators of the raster map. This is potentially convenient for drawing a spatial sample (see section 4) and for spatial statistical calculations. `ras2dta` allows maps to be imported with different spatial structures. The corresponding header information of each map is then saved as a separate file for each of the imported raster maps.

3.1 Syntax

`ras2dta , files(`*filelist*`)` [`idcell(`*varname*`) xcoord(#) ycoord(#) missing(#)`
`dropmiss extension(`*string*`) genxcoord(`*varname*`) genycoord(`*varname*`)`
`header saving(`*filelist*`) replace clear`]

3.2 Options

`files(`*filelist*`)` is required. It specifies the names of the ASCII files in *filelist* to be converted into Stata format. ASCII files must be located in the same directory and be listed with a separating space, without the file extension.

`idcell(`*varname*`)` is optional. It generates a spatial identifier (unique ID code) for the grid cells imported. The ID code starts at one in the top-left corner and increments in steps of one until the last cell at the bottom-right corner, from left to right and then top to bottom. `idcell()` saves the variable under the specified name in *varname*.

`xcoord(#)` is required if no header is present in the ASCII file. The number of X-values (number of columns or ncols) must be entered as an integer value.

`ycoord(#)` is required if no header is present in the ASCII file. The number of Y-values (number of rows or nrows) must be entered as an integer value.

`missing(#)` is optional. It must be specified if missing values are not the default ESRI no-data value of -9999.

`dropmiss` is optional. If specified, all default ($= -9999$) or user-defined (via `missing()`) missing values are dropped, including the corresponding codes from `idcell()`.

`extension(`*string*`)` specifies the file extension of the ASCII file. `extension(.asc)` is the default. For files without an extension, `extension(" ")` must be entered.

`genxcoord(`*varname*`)` is optional. It creates the variable *varname* carrying identifiers for the columns of the entire imported grid. X-coordinates will start with 1 at the top-left corner and increment to the right in steps of 1 (this is not affected by `dropmiss`).

`genycoord(`*varname*`)` is optional. It creates the variable *varname* carrying identifiers for the rows of the entire imported grid. Y-coordinates will start with 1 at the top-left corner and increment to the bottom in steps of 1 (this is not affected by `dropmiss`).

`header(`*filename*`)` optionally saves one ASCII header as a Stata data file for each imported grid. The header files are named `h_`*filename*, where *filename* is the name of the imported grid, with one variable called `hdr`. Existing files with the same name will be replaced.

`saving(`*filelist*`)` saves the Stata files under different names, as specified in *filelist* inside the parentheses of `saving()` (without separating comma and file extension). `saving()` will always save the files as Stata datasets. If `saving()` is specified, the number of imported grids specified in `files()` must equal the number of files specified in `saving()`. The default is to save the file in the same directory and under the same name as the original ASCII grid.

`replace` replaces existing files with the same name in the current working directory.

`clear` clears the data currently in memory.

4 Spatial sampling

The existence of spatial relationships among observations can result in unreliable estimates and misguided statistical inference of the parameters. Econometric problems

with spatial data can be due to interactions among neighboring agents. Spatial effects can also emerge when data from different sources, different sample designs, or varying aggregation rules is used (Anselin 1988).

One ad hoc technique to correct for spatial effects is to draw a systematic spatial sample from a grid. With systematic spatial sampling, a number of cells are selected in a regular fashion. This is done by keeping only cells that are a specified distance away from the nearest selected neighbor, resulting in a noncontiguous subsample of the data. Systematic spatial sampling permits the application of standard estimation methods (Anselin 2001).

The program `spatsam` draws such systematic spatial samples with a user-specified gap in the X- and Y-direction and optionally saves the sample as a new dataset. `spatsam` depends on the presence of X- and Y-coordinates that can be generated when importing the grids using `ras2dta`.

4.1 Syntax

`spatsam , gap(#) xcoord(`*varname*`) ycoord(`*varname*`) [insample(`*varname*`) norestore saving(`*filename*`) replace]`

4.2 Options

`gap(#)` is required and specifies the spatial lag between selected observations. For example, `gap(4)` specifies the selection of every fourth cell in the X- and Y-direction. Then the first observation in the sample is in the fourth row and fourth column, the second observation in the eighth row and fourth column, etc.

`xcoord(`*varname*`)` is required and specifies the variable that carries the X-coordinates.

`ycoord(`*varname*`)` is required and specifies the variable that carries the Y-coordinates.

`insample(`*varname*`)` is optional and saves the selected observations as a binary variable named *varname*. Selected observations get the value 1, and nonselected observations get the value 0. `insample()` is not affected by the use of `norestore`.

`norestore` prevents the restoration of the data previously in memory.

`saving(`*filename*`)` is optional and saves the data file under the name specified in *filename*.

`replace` replaces existing files with the same name in the current working directory.

For large raster maps, drawing a spatial sample with `spatsam` has the additional advantage that it significantly decreases the number of observation, thereby reducing the computational time.

5 Exporting Stata variables as raster grids

After importing data, managing data, making econometric estimations, and performing postestimation commands within Stata, the results may often be exported back to the GIS software package for a visual assessment and further spatial calculations. The program `dta2ras` takes Stata variables and saves them as ASCII raster grids in a format readable by most standard GIS software packages. If no *varlist* is specified, all the variables in the dataset are exported. The ASCII grid files include a standard header and can be readily imported into, e.g., ArcView (assuming Spatial Analyst or 3D-Analyst is loaded) with **File → Import Data Source → ASCII Raster**.

`dta2ras` asserts that the number of rows times the number of columns is equal to the number of observations. With one call of `dta2ras`, only variables of the same spatial structure (same number of rows and columns and the same cellsize) can be exported. If a spatial sample was previously drawn, `dta2ras` optionally expands the spatially sampled number of observations to the full grid size, i.e., to the number of rows times columns. The expansion requires the presence of a spatial identifier (`idcell()`) as generated by `ras2dta`. Missing identifiers in `idcell()` are filled with missing values to arrive at the desired total number of observations. After sorting by `idcell()`, each variable has the same structure and number of observations as in the original grid. This allows importing the variable back into the GIS software by inserting a previously saved header file or by manually providing the spatial structure of the raster grid with at least the information on columns and rows in `xcoord()` and `ycoord()`, and (optionally) also with the remaining information on `cellsize()`, `xllcorner()`, `yllcorner()`, and `missing()`.

5.1 Syntax

`dta2ras` *varlist* `,` { `header(`*filename*`)` | `xcoord(`#`)` `ycoord(`#`)` } [`cellsize(`#`)`
`xllcorner(`#`)` `yllcorner(`#`)` `missing(`#`)` `idcell(`*varname*`)`
`idfile(`*filename*`)` `expand` `norestore` `saving(`*filelist*`)` `replace`]

5.2 Options

`header(`*filename*`)` specifies the header file, which must be a Stata data file named *filename* with one variable named `hdr`. `ras2dta` can automatically create this file. If this option is not specified, `xcoord()` and `ycoord()` are required.

`xcoord(`#`)` is required if `header()` is not specified and defines the number of X-coordinates (columns or ncols) as integer values.

`ycoord(`#`)` is required if `header()` is not specified and defines the number of Y-coordinates (columns or ncols) as integer values.

`cellsize(`#`)` is optional. It specifies the cell size of the resulting grids. The default is `cellsize(1)`.

`xllcorner(#)` is optional. It specifies the X-coordinate of the lower-left cell. The default is `xllcorner(1)`.

`yllcorner(#)` is optional. It specifies the Y-coordinate of the lower-left cell. The default is `yllcorner(1)`.

`missing(#)` is optional. It must be specified if missing values are *not* the default ArcInfo/ArcView no-data value of −9999. The default is `missing(-9999)`.

`idcell(`*varname*`)` is a variable carrying the spatial identifier (ID code) of the grid cells and is required if `expand` is specified without `idfile()`. The upper-left cell in `idcell()` starts at 1 and must increment in steps of 1 moving from left to right, and then top to bottom.

`idfile(`*filename*`)` is the Stata data file that carries the spatial identifier (ID code) of the grid cells and is required if `expand` is specified without `idcell()`. The upper-left cell in `idfile()` must carry the identifier 1 and must increment in steps of 1 moving from left to right, and then top to bottom. If `idfile()` is specified, the identifying variable in the master and using data must have the same name.

`expand` expands the dataset to the full number of observations, e.g., if a spatial sample was previously drawn using `spatsam`. `expand` depends on the presence of `idcell()` or `idfile()`.

`norestore` prevents the restoration of the data previously in memory.

`saving(`*filelist*`)` saves the ASCII files under different names, as specified in `filelist` (names separated by spaces, without comma and file extension). `saving()` saves the files in ASCII format. If `saving()` is specified, the number of exported variables in *varlist* must equal the number of files specified in *filelist*. The default is to save the raster grids under the exported variable names with the ending `_o.asc`.

`replace` replaces already existing files of same name in the current working directory.

6 A numerical example

A subset of real-world data from the Central Highlands of Vietnam is used as a numerical example for econometric estimation with spatially explicit raster data. The data is described in Müller (2003) and stems from a research project on the determinants of land-use change. Forest cover (`forest`) is chosen for the purposes of this paper as the binary dependent variable (forest = 1 and nonforest = 0). As covariates, several indicators describing the agricultural potential of each raster cell are used, comprising of slope (`slp`), elevation (`elev`), soil suitability (`soil`), as well as the Euclidean distance to major roads (`disroad`) as a proxy indicating access to markets and transportation costs. For simplicity reasons, I omitted a range of other variables that potentially influence forest cover, such as population density, the introduction of technologies, or the occurrence of protected areas. The dependent variable and the four covariates are calculated and stored within a GIS as raster layers with the same geographic projection, grid cell size, and spatial extent.

6.1 Data preparation

From within the GIS software, each raster layer is exported as an ASCII data file. The layer forest cover as the dependent variable `forest` is imported to Stata with

```
. ras2dta, f(forest) header idcell(idc) genx(x) geny(y) replace clear
```

```
No of columns:             389
No of rows:                347
number of cells:       134,983
file forest.dta saved
```

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
idc	134983	67492	38966.38	1	134983
forest	134983	.8371795	.3692032	0	1
x	134983	195	112.2947	1	389
y	134983	174	100.1702	1	347

The variable `idc` carries a unique identifier for each observation. The binary variable `forest` indicates a forest cover of 84%. Column values are contained in the variable `x`, advancing from column one to column 389, and restart at one in the second row. This procedure is repeated 347 times for every row. Row numbers are named `y`, advancing from row one to row 347 in blocks of 389 (columns). The spatial information of the raster grid for `forest` is saved in `h_forest.dta`:

```
. use h_forest, clear
. list, sep(2)
```

	hdr
1.	ncols
2.	389
3.	nrows
4.	347
5.	xllcorner
6.	223331.63634
7.	yllcorner
8.	1372962.4841
9.	cellsize
10.	50
11.	NODATA_value
12.	-9999

Observation two specifies the number of columns (389), and observation four specifies the number of rows (347). The total number of observations is the product of columns and rows (134,983). The lower-left coordinate, referenced to a projected coordinate system (in this case, to Universal Transverse Mercator, UTM) is represented by observation six

x and eight y, respectively. The map units of the raster grids are specified in meters within the GIS. Therefore, the size of a square grid cell is 50 by 50 meters. The total area covered by the map is the number of observations (134,983) times the area of one cell (2,500 m^2) or approximately 337.5 km^2.

The four raster maps used as covariates are imported using `ras2dta`. A variable carrying the spatial identifier (`idcell()`) and X- and Y-coordinates are generated with the options `idcell(idc)`, `genxcoord(x)`, and `genycoord(y)` for each of the four layers.

```
. foreach grid in elev slp soil disroad {
  2.         qui ras2dta, f(‘grid’) idcell(idc) header genx(x) geny(y) drop
> replace clear
  3.         qui use h_‘grid’, clear
  4.         cf hdr using h_forest
  5. }
```

The built-in Stata command `cf` verifies that the headers of all the imported rasters are identical. We now `merge` the covariates based on the spatial identifier, the X- and the Y-coordinates to the dependent variable `forest` with observations uniquely identified in both datasets. The complete dataset is saved as `forestfull.dta` for later use in postestimation commands.

```
. use forest, clear
. merge idc x y using elev slp soil disroad, unique
. drop _merge*
. lab def for 0 "Non-forest" 1 "Forest"
. lab val forest for
. lab var forest "Observed"
. save forestfull, replace
file forestfull.dta saved
```

The spatial sample is drawn by selecting every third cell in the X- and Y-direction.

```
. spatsam, gap(3) xcoord(x) ycoord(y) saving(forsam3) replace
you selected 14835 observations
file forsam3.dta saved
```

And another sample with a gap of five cells to allow for the comparison of estimation results.

```
. spatsam, gap(5) xcoord(x) ycoord(y) norest
you selected 5313 observations
```

With a gap of five cells in the X- and Y-direction, every cell is 200 meters (four cells of 50 meters each) away from its nearest neighbors in all four directions. This sampling procedure reduced the number of observations from almost 135,000 to 5,313. The gap of three cells resulted in 14,835 observations, with every cell being 100 meters from the nearest neighbors.

6.2 Econometric estimation

The resulting dataset we use to fit standard binary logit models. First, for the gap of five cells:

```
. logit forest slp elev soil disroad, nolog
Logit estimates                                   Number of obs   =       5313
                                                  LR chi2(4)      =    2995.12
                                                  Prob > chi2     =     0.0000
Log likelihood = -877.67432                       Pseudo R2       =     0.6305

------------------------------------------------------------------------------
      forest |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         slp |   .1217579   .0156945     7.76   0.000     .0909972    .1525185
        elev |   .0465324   .0032897    14.15   0.000     .0400847     .05298
        soil |  -.1377835   .0625698    -2.20   0.028     -.260418    -.015149
     disroad |   .5697638   .0654302     8.71   0.000      .441523    .6980045
       _cons |  -22.68067   1.470205   -15.43   0.000    -25.56222   -19.79912
------------------------------------------------------------------------------
note: 0 failures and 1466 successes completely determined.
```

And, second, for the gap of three cells:

```
. use forsam3, clear
. logit forest slp elev soil disroad, nolog
Logit estimates                                   Number of obs   =      14835
                                                  LR chi2(4)      =    8302.53
                                                  Prob > chi2     =     0.0000
Log likelihood = -2455.8097                       Pseudo R2       =     0.6283

------------------------------------------------------------------------------
      forest |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         slp |    .10766   .0090433    11.90   0.000     .0899355    .1253846
        elev |    .047577   .0020051    23.73   0.000     .0436472    .0515069
        soil |  -.1796031   .0378277    -4.75   0.000    -.2537439   -.1054622
     disroad |   .5921515   .0393804    15.04   0.000     .5149674    .6693356
       _cons |  -23.02267   .8948099   -25.73   0.000    -24.77647   -21.26887
------------------------------------------------------------------------------
note: 0 failures and 4155 successes completely determined.
```

The variables have the expected signs and significantly influence the probability of observing forest cover. Cells with higher slope and elevation, with lower soil suitability for agriculture, and that are further away from main roads are more likely to be under forest cover. The results from the estimation with the gap of three cells and with the gap of five cells differ, as expected, in coefficient size, standard errors, significance levels, and confidence intervals. However, the coefficients still have the expected signs and are all significant at the 1% and 5% levels, respectively. Therefore, the substantive findings from the regressions can be retained unchanged.

Predicted probabilities facilitate meaningful interpretations. To obtain the predicted probabilities and to observe the prediction accuracy, we compute the predicted probability of a positive outcome for each cell given the values of the covariates for that observation and the number of correctly predicted cells. The higher the predicted prob-

abilities, the higher is the likelihood that a cell is under the predicted outcome "forest cover". Prediction values for the estimated probabilities are created by using the highest predicted probability as the predicted outcome, i.e., probability values exceeding 0.5 in the case of binary dependent variables. This choice is arbitrary, and alternatively, higher thresholds can be used to assess the prediction accuracy and prediction strength. Increasing prediction thresholds will decrease the number of correctly predicted cells.

The estimated parameters of the reduced dataset are derived following the spatial sample with a gap of three cells. The predicted probabilities are computed from these parameters for all observations at every location of the study area to generate a continuous surface that allows a more accurate quantitative and visual postestimation assessment.

```
. use forestfull, clear
. predict pforest, p
. gen predfor = 0
. replace predfor = 1 if pforest > 0.5
(111251 real changes made)
. lab var predfor "Predicted"
. lab val predfor for
. gen predcorr = (forest == predfor)
. summarize pforest forest predfor predcorr
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pforest	134983	.8373526	.2884086	.0497403	1
forest	134983	.8371795	.3692032	0	1
predfor	134983	.8241853	.3806639	0	1
predcorr	134983	.9302283	.2547629	0	1

93% of the cells are predicted correctly using the highest predictions as the predicted values. Another measure of goodness of fit is the prediction matrix where actual values are tabulated against the predicted values. The prediction matrix facilitates the assessment of model accuracy as a summary of the model fit, as it reduces the model complexity. Correct predictions are shown in the diagonal of the prediction matrix:

```
. tab forest predfor
```

	Predicted		
Observed	Non-fores	Forest	Total
Non-forest	18,146	3,832	21,978
Forest	5,586	107,419	113,005
Total	23,732	111,251	134,983

5,586 locations under forest cover are predicted to be nonforest, while 3,832 locations are wrongly predicted as forest. 125,565 (18,146 + 107,419) observation or 93% (see above) are predicted correctly.

6.3 Export of results and geovisualization

The prediction matrix yields no insights into the spatial accuracy of the predictions. The tabulation of predicted probabilities can also be examined graphically in prediction maps or mapped probability values (Nelson and Geoghegan 2002). To do this, the predictions are exported to an ASCII raster grid:

```
. dta2ras pforest, header(h_forest) expand idcell(idc) replace
(note: file pforest_o.asc not found)
pforest saved as -pforest_o.asc-
```

The ASCII file `pforest_o.asc` is imported into the GIS software package to map the predicted probabilities. The resulting prediction map in figure 2 visualizes the probabilities to observe forest cover, generated from the variable `pforest`.

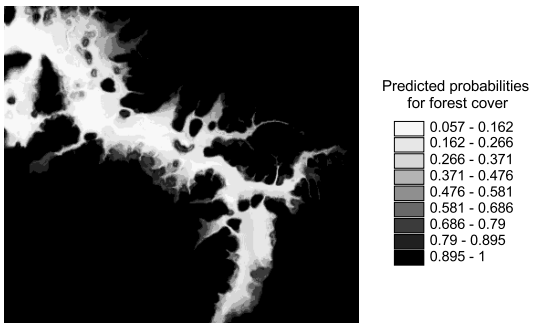

Figure 2: Predicted probabilities

The values of predicted forest cover, `predfor`, are converted in the same way into an ASCII raster file:

```
. dta2ras predfor, header(h_forest) expand idcell(idc) replace
(note: file predfor_o.asc not found)
predfor saved as -predfor_o.asc-
```

The geovisualizations in figures 3 (a) and (b) give a good indication of the prediction accuracy and facilitate visual analysis of the spatial accuracy. Overlaying the actual forest cover in figure 3 (a) with the predicted forest cover in figure 3 (b) allows for a rapid assessment of the quantity and location of incorrect predictions. Further it helps to identify potential hot spots, or areas of risk, for future deforestation and can be of great policy relevance for development planning and resource management efforts.

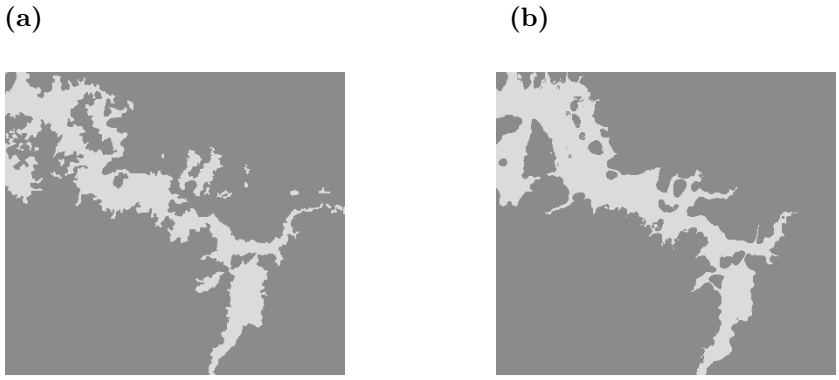

Figure 3: Observed (a) and predicted (b) forest cover (in darker gray)

7 Acknowledgments

The research project from which this work evolved was financed by the Tropical Ecology Support Programme (TÖB) of the German Technical Cooperation (GTZ) with a grant from the Federal German Ministry for Economic Cooperation and Development (BMZ). The writing of the paper was supported by the Deutsche Forschungsgemeinschaft (DFG) under the Emmy Noether Program. I am grateful for the generous help of Nicholas J. Cox for programming support and the useful comments from Ronnie Babigumira and one anonymous reviewer. Any remaining errors are mine.

8 References

Anselin, L. 1988. *Spatial Econometrics: Methods and Models*. Dordrecht: Kluwer Academic Publishers.

—. 2001. Spatial Econometrics. In *Companion to Theoretical Econometrics*, ed. B. H. Baltagi, 310–330. Oxford: Blackwell Scientific Publications.

Environmental Systems Research Institute (ESRI). 2000. *ArcView 3.2 On-line Help*. Redlands, CA: Environmental Systems Research Institute, Inc.

Müller, D. 2003. *Land-use Change in the Central Highlands of Vietnam: A spatial econometric model combining satellite imagery and village survey data*. Doctoral dissertation, Institute of Rural Development, Georg-August University Göttingen: http://webdoc.sub.gwdg.de/diss/2003/mueller/.

Nelson, G. C. and J. Geoghegan. 2002. Deforestation and land use change: sparse data environments. *Agricultural Economics* 27(3): 201–216.

Tobler, W. 1979. Cellular Geography. In *Philosophy in Geography*, ed. S. Gale and G. Olsson, 379–389. Dordrecht: Reidel.

About the Author

Daniel Müller is an agricultural economist presently affiliated as a postdoctoral researcher in a Junior Research Group on Postsocialist Land Relations at the Humboldt University of Berlin.