Integrating spatial data on the rural economy, land use and biodiversity
RELU Development activity
RES-224-25-0099

PI: Dr Piran White, Environment Department, University of York Heslington, York, YO10 5DD, UK. E-mail: pclw@york.ac.uk

http://www.relu.ac.uk/research/People&RuralEnvironmentDA.htm

Description of Land Use Modelling Methods

Original report written by Dr Andrew Crowe, Lancaster Environment Centre, Lancaster, LA1 4YQ,UK. E-mail: a.crowe@lancaster.ac.uk
Edited by Dr Colin McClean, Environment Department, University of York, Heslington, York, YO10 5DD, UK. E-mail: cjm8@york.ac.uk

Data Sets

Agricultural Census

The UK agricultural census has been performed by UK government departments since 1866, the responsibility now falling to the Department for the Environment, Farming and Rural Affairs (DEFRA) (http://www.defra.gov.uk/esg/work_htm/publications/cs/farmstats_web/default.htm). Each year approximately 60% of agricultural holdings in the UK receive a postal questionnaire on how much and what type of land use and livestock is associated with the holding along with the amount of labour employed on the holding. Up to 2004 the geographical delineation for the regional statistics have been based on the Eurostat Nomenclature of Units for Territorial Statistics (NUTS) as defined by the Ordinance Survey ward boundaries. From 2004 onwards this was changed to Office of National Statistics super output areas (SOA). For census data up to and including 2003 the lowest level of data collation publicly available was at the ward (NUTS 5) level.

The agricultural census contains the areas under 26 crops which can be grouped into the classes listed in Table1. Data in the agricultural census may be suppressed if the information can be used to identify one or two holdings within the summary region. With the widespread land use types this suppression is not a common occurrence, as most holdings will have some land under that particular land use. However, for minor land uses suppression of the land use area data can be relatively common. In some cases this suppression of one or two land uses leads to the total area under agricultural management to also be suppressed. The wards where total area had been suppressed were removed from the analysis.

Table 1. Agricultural census items making up the eight main land use classes

Land Use Class	Agricultural census items
Grazing	Temporary Grassland; Permanent Grassland; Rough Grazing
Arable	Wheat; Winter Barley; Spring Barley; Oats; Maize; Other Cereal; Other Arable
Root Vegetables	Potatoes; Sugar Beet; Turnips
Other Vegetables	Field Beans; Dry Harvested Peas; Peas and Beans; All Other Vegetables and Salad; Other Crops for Stockfeed
Oilseed	Oilseed Rape; Linseed
Non-farmed	Bare Fallow
Miscellaneous	Horticulture; Under Glass; Top Fruit; Small Fruit; Hardy Nursery; Bulbs and Flowers
Unclassified	Unclassified

In order to fill in the suppressed land use data within a ward, regional level (NUTS Level 1) statistics were used to assign appropriate proportions of the total suppressed area to each of the suppressed land uses within a ward. The regional land use data was converted to a set of proportions based on the total area under crops and agricultural grassland. For each ward, the regional proportions were rescaled so that the proportions for the land uses suppressed at the ward level summed to one. The total suppressed area was divided between the suppressed land uses according to the rescaled regional proportions. In some cases land use areas at a regional level were also suppressed and a similar procedure was applied using the national level statistics. The reported total area for the regions was typically larger than the sum of the land use classes within that region. For the regions that required filling, the total area was reduced by multiplying by a factor estimated from the regions where a complete dataset was presented. It is the reduced total that is used in calculating the suppressed land use areas.

Land Cover

The Land Cover Map of Great Britain 2000 (Fuller et al.2002) was used: http://www.ceh.ac.uk/sections/seo/lcm2000_home.html. The 1 km^2 summarised land cover map was downloaded from the Countryside Information System web site http://www.cs2000.org.uk/CIS_files.htm , giving1 km^2 grids for ten land cover classes (Table 2). Each cell in a grid records the estimated area of that land cover within each of the 1 km^2 grid cell.

Table 2. Land Cover Classes

Land Cover Class
Broadleaf Woodland
Coniferous Woodland
Arable
Improved Grazing
Semi-natural Grassland
Open Water
Upland
Built-up Areas
Coastal
Sea

Linking the datasets

The land use and land cover data require to be referenced to the same units. The land cover grids were loaded into ArcGIS along with a shapefile containing the ward boundaries. The areas of each land cover in the ward were calculated by adding the values for grid cells present within a ward. For those cells that are intersected by the boundary of the ward, the value for that cell was scaled by the proportion of the cell occurring in the ward.

The area of the ward was in most cases greater than the total area reported under agricultural land uses. Therefore an addition land use category, designated as unclassified land use, was added to store the difference between the agricultural land uses and the total area of the ward. In 438 wards, the total area reported under agricultural land use exceeded the total area of the ward. In these cases the areas of agricultural land use was rescaled to bring the total area in agricultural land use to the same value as the ward area. For the genetic algorithm the values for land cover and land use were re-expressed as proportions of the ward area

Methods

Genetic Algorithms

Background
Genetic algorithms are heuristic optimisation techniques that are based on ideas borrowed from evolutionary biology. The three main components of the genetic algorithm process are survival of the fittest, recombination and mutation (Beasley *et al*. 1993). Survival of the fittest is implemented by allowing better solutions to the problem to be selected for the new generation more often than poorer solutions. Recombination occurs by swapping values between solutions and mutation by changing the values within a solution.

Solutions to the problem are coded as a series of values in a structure referred to as the chromosome. At the start of the genetic algorithm process, a number of random

solutions are created and stored in the chromosomes. All the chromosomes are evaluated and given a fitness value based on how good the encoded solution is. These values are then used to populate the next generation of chromosomes so that fitter chromosomes represent a larger proportion of the copied chromosomes than the less fit solutions. These are then subjected to recombination and mutation to create new combinations of values that can then be re-evaluated. This process is repeated for a large number of generations with the goal being that the solutions get better with each successive generation but with the ability to escape local fitness maxima in order to find the best overall solution.

Use in this context

In this study the genetic algorithms are used to find a relationship between land use and land cover classes. The use of the GA method can be justified by the problems associated with attempting to make such a relationship. The main issue in this study is that the relationship between a particular land cover and land use is expected to vary according to the composition. This makes the optimisation problem non-smooth.This rules out a simple regression analysis, as the relationship for between land cover and land use must be determined for all land cover classes simultaneously.

As the matrix of land uses by land covers is large, the problem would be very difficult if not impossible to solve using analytical optimisation methods. Genetic algorithms have been shown to be strong candidate methods in studies where the system does not lend itself to being mathematically characterised.

Land Use Simulation Model

In this study we are attempting to use a land cover map to guide the spatial allocation of land use. To do this we apply the probabilities from the genetic algorithm analysis to the 1 km grid of land cover. This is a static allocation based purely on the relationship defined by the genetic algorithm. It assumes that the relationship is constant over the entire area. As we know the area under each land use and land cover, it is the relative relationship between the probabilities associated with land cover/land use pairs that define the final spatial allocation rather than the values themselves. The simulation is stochastic, so each run of the simulation will yield a different spatial pattern of land use. However the differences between runs are expected to be small, with the general pattern of cells with high or low area attributed to a particular land use maintained across runs.

Why use the python/mysql/c++ approach?

The method developed for this study uses a combination of a database to store and manipulate the data, a standalone application to perform the analysis and simulation and a scripting language to coordinate the analysis. The applications used represent examples of free software with high levels of support.

Python is an object orientated scripting language with a General Public License (GPL) and a good support and development structure. Python was chosen as the scripting language mainly due to the ArcGIS (ESRI, 2004) scripting language moving from AML and VBA to Python.

The decision to hold the data on a database rather than leaving it in flat files allows a high level of flexibility in running the analysis. The MySQL database application was chosen, as this is available under a General Public Licence (GPL) and is well supported with GUI query and administration applications. There are also open database connectivity (ODBC) drivers allowing ArcGIS to connect to the MySQL database and a Python module, meaning the database can be queried from a Python script. Python could therefore be used to dynamically generate the files required to run the standalone genetic algorithm and simulation applications.

The python script also allows multiple runs to be performed without the need for the user to create new files or change parameters by hand. The standalone programs have been written in C++ and compiled on the Dev C++ IDE, again available free under the GNU license and based on a windows port of the GCC compiler. Development on the GCC compiler minimises the problems associated with porting the source code between operating systems.

Detail of GA
Chromosome structure

In this study the GA chromosomes are comprised of 10 sets of 26 values. These represent the proportions of the 26 land uses in each of the 10 land cover classes. In this case the values for the land uses within a land cover must always sum to unity, a condition that is set when the initial chromosome is created. To make sure the summation to unity is maintained, recombination is restricted to points between land cover sets rather than at a random position within the sets and mutation is performed by transferring part of the proportion attributed to one land use to another land use within the same set. The fitness function used to evaluate the chromosomes is the sum of absolute residuals.

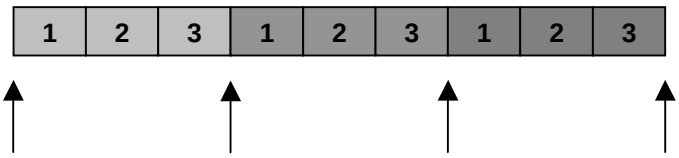

Figure 1. Diagrammatic representation of a chromosome containing three land uses (1,2,3) for three land cover types. Arrows indicate locations where crossover would be allowed to occur.

The optimisation routine
The selection of chromosomes for the new generation is performed by tournament in which two chromosomes are selected at random from the current generation and the chromosome with the lower fitness value copied to the new generation. The selection occurs with replacement so a chromosome can be selected multiple times. In addition to this the two chromosomes with the lowest sum of absolute residuals are automatically copied into the new generation. These two chromosomes are not subjected to recombination or mutation so that the best solution is never lost from the

analysis. The procedure of copying the best chromosomes into the next generation and leaving the values unchanged is referred to as elitism.

Running the GA

A python script file was used to create a set of files containing the land use proportions for wards in each region for the years of 2000, 2001, 2002 and 2003. A similar script was used to create the files containing the land cover proportions for the wards on each region. Python scripts were used to run the genetic algorithm application for each region across the four years, and to run the genetic algorithm against the data for Yorkshire and Humberside region an additional 9 times to investigate the variation in the GA estimates across differing runs. The genetic algorithm application was set to run with 30 chromosomes per generation over a maximum number of 1500 generations.

Detail of the Simulation
How the simulation assigns land use to land cover

The spatial simulation assigns the land use in two stages. The first stage uses the probabilities from the genetic algorithm to assign land cover/land use pairs within a ward. The simulation creates two lists, one from the land use areas and the second from the land cover areas. Each list contains an item with the appropriate code for every unit area of land use or land cover. The simulation assigns land cover/land use pairs by picking a random land use followed by a random land cover from the lists of unassigned land cover and land use. It then compares the probability value for this land cover/land use pair to a randomly selected value between 0 and 1. If the random value is less than or equal to the probability of the land cover/land use pair, the pair is added to a list of assigned pairs and the items removed from the lists of unassigned land use and land cover.

If the value of the random number is greater than the probability, a new land cover item is selected from the list and a new random value generated. The new value is again tested against the probability. This process continues until the land use is successfully assigned to a land cover. Once the land cover/land use pair is assigned, the simulation moves on to pick new random land use and land cover items from the list until all the land uses are paired with land covers.

The second stage of the simulation involves assigning the land cover/land use pairs to the 1 km grid cells. A list of land cover/grid cell identity values is created. The simulation then assigns a land use to each land cover/grid cell identity pair by selecting a random land cover/land use pair with the corresponding land cover from the list created in the first stage of the simulation. The simulation then totals up each of the land uses within each grid cell and outputs them to a text file with rows representing the grid cells and columns representing land uses.

Running the simulation through Python/MySQL

As with the genetic algorithm, a Python script was used to dynamically generate the files required by the simulation application from data stored in the MySQL database.

The script extracts a list of wards with their associated regions from the database. Ward by ward, the script queries the database to extract the land use area, land cover area and a list of the 1 km grid cells that are partially or full covered by the ward. From this query the script creates the land use, land cover and cell identity files which are supplied to the simulation application. Once the simulation has run, the script loads the resulting file back into a database table. When the simulation has been completed for all the wards, a query is run against the results table to sum the land uses within each 1km grid cell.

References

Beasley, D, Bull, D.R.& Martin, R.R. (1993) An overview of genetic algorithms. 1. Fundamentals. *University Computing*, 15 (2): 58-69 1993.

Fuller, R.M., Smith, G.M, Sanderson J.M., Hill R.A. & Thomson A.G. (2002) The UK Land Cover Map 2000: Construction of a parcel-based vector map from satellite images. *Cartographic Journal* 39 (1): 15-25.