AUTOLOGIT.CC - AN IMPLEMENTATION OF THE AUTOLOGISTIC MODEL WITH COVARIATES

Greg S. Young*
National Center for Atmospheric Research, Boulder, Colorado
and
Jennifer A. Hoeting
Colorado State University, Fort Collins, Colorado

The C++ program **autologit** produces predictions for an autologistic model with covariates using the methodology described in Hoeting *et al.* (2000). An object-oriented design methodology was adopted for implementation. A Gibbs sampling estimation procedure is used to estimate the parameters. This software will be made publicly available at Statlib (http://lib.stat.cmu.edu).

The file **AUTOLOGIT.shar** contains the following files:

- **autologit.cc**- main program. This program is actually called by the user and user modifications should be limited to this program. Several subroutines are called internally and are outlined in the header file **autologit.h**.
- **param.cc** - object file. This file contains the member functions for the **param** class (see below). All parameter manipulations are managed by these routines, including likelihood optimization.
- **stats.cc** - object file. Routines for the generation of random variates are included here, written by Dixon (1998).
- **matrix.cc** - object file. All of the procedures for the needed matrix manipulations are included here.
- **gridpoint.cc** - object file. This file contains the the member functions for the gridpoint class.
- **Makefile** - project makefile. This routine will compile the object files and link them to the main program in the correct order, building the executable file **autologit**.

In addition, a variety of header (.h) files are included, specifying the procedures and class structures.

As currently written, **autologit** computes estimates for square lattices. A second order neighborhood is used in the computation of the spatial covariate (β). See the procedure "get_sums" in **autologit.cc** to change the neighborhood structure.

1 Classes

In an object oriented design, classes include a set of data and all of the operations that will be performed on those data in a single structure. The goal is to make code that is simpler and easier to maintain. The following classes were designed:

* *Corresponding author address*: Greg Young, National Center for Atmospheric Research, Research Applications Program, Boulder, CO 80307-3000; email: young@ucar.edu

- **param** - a parameter base class, containing the information necessary for the assignment and updating of the parameter values. The classes beta_param, theta0_param, and thetaj_param are derived from the class param, and correspond to the parameters β, θ_0, and θ_k (for $k = 1, 2, ..., p$) respectively. The primary reason for the distinction was the differing pseudolikelihoods of the parameters. There are no instances of the param class, only the derived classes. While there can be only one instance of beta_param and theta0_param, multiple objects of the thetaj_param class are allowed to correspond to the number of covariates. The procedure used in the optimization of the parameter pseudolikelihoods was programmed by Keselyov (1991), based on the algorithm by Forsythe *et al.* (1977).

- **gridpoint** - a class describing the status of each grid point on the lattice. The most significant member function updates the grid point's probability of presence.

- **matrix** - a matrix class. This class resulted from a need to perform copious numbers of matrix manipulations. Although the class is somewhat limited in its scope, all necessary functions are included and are straightforward enough for easy modification.

2 Installation

A **Makefile** has been included for the compiling of the program and its various object files. Simply copy all of the files into one directory, and type "make". The program will be complied and the executable **autologit** will be created. The g++ complier is used. If any modifications are made to any of the included programs, rebuild the code by typing "make clean" (this removes the old object files) and "make" (to recompile the code). A debug option is also included in the **Makefile**, but must be specified in a build by typing "make debug".

3 Usage

The following sections detail how to set up the necessary files for **autologit** and how to customize its features.

3.1 Reading in arrays

Much of the data used in **autologit** is in the form of arrays or vectors. As described in Hoeting *et al* (2000) section 3, the image ($\underline{x}$) is updated in four independent sets. These sets must be manually coded and are then read into the program. Also, the response vector ($\underline{y}$) and the list of sampled sites are in array form.

The format for reading in these arrays is from an ASCII text file, where the first line is the number of elements in the array, followed by the actual elements, listed one per line. For example, the vector $(1, 0, 1, 1)$ would be specified as

```
4
1
0
1
1
```

in a file. If this vector was a reponse, it would translate into the following 2 X 2 lattice

$$\begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix}.$$

The format for the response vector is 0 for a "no" observation or for an unsampled site, and 1 for a "yes" observation. There must be one response for every site in the lattice. That is, the vector must specify a 0 or a 1 for all of the sites. The length of the vector will equal the number of sites. The vector of sampled sites must also be equal to the number of sites in the list with a 0 representing a sampled site and a 1 representing an unsampled site. This format may seem backward, but it aides in the updating of the sites in the sampler.

The four independent groups are identified in the following manner. Arrays must be read in with the following file names according to their location (top, bottom, left, right, or middle) and group number (1, 2, 3, 4): bot3, bot4, left1, left3, right2, right4, top1, top2, mid1, mid2, mid3, mid4. The non-middle arrays have only two separate files because as edges, they will contain only two of the four groups. The files contain on the first line the length of the array and then the index numbers of the sites included in that group. For example, for the top edge of group 2 the following would be the contents of the file **top2** for a 6 X 6 lattice:

```
3
2
4
6.
```

A site can only occur in one group. For corner sites, choose one of the two possibilities.

These groups have already been specified for a 50 X 50 lattice. Use them as guidelines for constructing these groups for different lattices.

3.2 Reading in the covariate matrix

The covariate matrix contains one row for each site and one column for each covariate. There cannot be any sites with missing covariates. The matrix is described in a similar manner to the arrays. The first line of the file contains the number of rows in the matrix followed by the number of columns. Then, the matrix values are listed, one per line, column by column. For example, the matrix

$$\begin{vmatrix} 1 & 5 \\ 2 & 8 \\ 12 & 3 \end{vmatrix}$$

would specified

```
3 2
1
2
12
5
8
3.
```

3.3 Customizing

There are several customizable features in the **autologit** program. The file **settings.h** includes three constants that need to be set by the user. The first is SIZE which determines the size of the lattice used. Only squared lattices are possible and SIZE determines the length and width of the lattice (in number of sites). The second is N_COVARS, which specifies the number of covariates that will be considered for the model. The third is the string **file_loc** which gives the directory of the files specifying the independent groups.

The remaining inputs are specified through command line arguments. When autologit is run, the following must be specified and in this order:

- response file
- covariate file
- sites sampled file
- simulation length (in number of iterations)
- burn-in time (in number of iterations)
- τ (the variance for the normal prior distributions of θ_k, $k = 0, ..., p$)
- ψ (the first parameter in the gamma prior for β where the mean is $\psi\alpha$ and the variance is $\psi\alpha^2$)
- α (the second parameter in the gamma prior for β)
- starting value for β
- starting values for θ_k, $k = 0, ..., p$ (this must be of length N_COVARS+1)
- directory for the output files.

For example, the command line

```
autologit inputs/y inputs/Z inputs/g 2000 500 20 1 2 1 0 0 0 .
```

would use the file `y` in the directory `inputs` for the response ($\underline{y}$), the file `Z` in the directory `inputs` for the covariate matrix, and `g` in the directory `inputs` for the vector of sampled sites. The simulation would run for 2000 iterations, allowing for a 500 iteration burn-in period. The variance for ($\underline{\theta}$) would be 20, ψ = 1, and $\alpha = 2$. The starting values would be $\beta = 1$, $\theta_0 = 0$, $\theta_1 = 0$, and $\theta_2 = 0$, where the constant N_COVARS was set to 2. Finally, the output files would be written to the directory from which the command originated (the current directory).

3.4 Output

Three files are output from **autologit**. The first is **autologit.avg** which is the average image or mean of the estimated posterior distribution for the probability of presence. This file stores the vector in the same orientation as the file for the input vector $\underline{y}$ and the vector of sampled sites. That is, the output file

```
.67
.35
.78
.20
```

.67	.78
.35	.20

would correspond to the following 2 X 2 estimated posterior probability lattice

The second is the file **track.p** stores every sim/10th estimated posterior. That is, it records 10 estimated posteriors, equally spaced over the course of the simulation run. The setup is similar to that of **autologit.avg** except that each of the 10 realizations are separate "columns" in the file (each line in the file corresponds to a grid point with commas separating the realizations).

Lastly, the file **track.param** stores the parameter estimates at every iteration. The estimates are listed one iteration per line, with the first entry being the estimate for θ_0, followed by the θ_k $(k = 1, ..., p)$ estimates, and finally the estimate for β.

References

Dixon, M. 1998: Random variate generation. University Corporation for Atmospheric Research.

Forsythe, G., M. Malcolm, and C. Moler, 1977 **Computer Methods for Mathematical Computations**. Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Hoeting, J. A., M. Leecaster, and D. Bowden, (2000): An improved model for spatially correlated binary responses. *Journal of Agricultural, Biological, and Environmental Statistics*, **5**, 102-114.

Keselyov, 1991: FMINBR. Obtained from Netlib repository.