Nearest Neighbor Imputation using the yaImpute package in R

Objectives

- Learn about nearest neighbor (NN) imputation
- Get an overview of the functions and features available in the yaImpute R package
- Impute two variables of interest simultaneously using different nearest neighbor imputation approaches

Background on nearest neighbor imputation

NN imputation methods were developed to fill in observations of variables that are missing on some records (*Y-variables*) using related variables that are available for all records (*X-variables*).

Donor-based method: Fill in missing values of *Y* with records having similar characteristics in *X*

Y-variables: expensive, sparse data only available on a sample of units

X-variables: spatially comprehensive, inexpensive data available for all units

Target record: database record with only X-variables measured

Reference record: database record with X- and Y-variables measured

Advantages:

- multivariate
- Retain complex variance-covariance structure and natural variation of the Y-variables as long as k=1
- Imputed Y-variables will be within the bounds of biological reality as long as k=1
- Do not extrapolate outside range of sampled conditions
- Do not rely on any underlying probability distribution for estimation → non-parametric or distribution-free
- Only assumption: X-variables have a strong relationship with Y-variables → complete knowledge of complicated relationships between X- and Y-variables not needed

Distance metrics:

Distance metrics determine similarity between target and reference records

Absolute differences: $d_{ij} = \sum_{l=1}^{p} c_l \left| x_{il} - x_{jl} \right|$

x_{il} and x_{jl} are the values of X-variable *l* for the target record *i* and reference record *j*, respectively, *p* is the number of X-variables, and c_l is the coefficient for variable x_l

Quadratic form used for Euclidean, Mahalanobis and other distance functions:

$$d_{ij}^2 = \left(x_i - x_j\right) W \left(x_i - x_j\right)^T$$

x_i and x_j are (1 x p) vectors of X-variables for the ith target and jth reference record, respectively, W is a (p x p)symmetric matrix of weights

Number of neighbors k:

either one single neighbor (k=1) is used as donor or a simple or weighted average of k > 1 near neighbors

optimal value of k is a trade-off between the accuracy of the estimates and the variation that is retained in the estimates

yaImpute package

tailored to imputation-based forest attribute estimation and mapping

Distance metrics:

yaImpute offers several methods for finding nearest neighbors (Table 1 in (Crookston and Finley 2008):

Method	Value of **W**
`raw`	Identity matrix, **I**
`euclidean`	Inverse of the direct sum of the X's covariance matrix
`mahalanobis`	Inverse of the X's covariance matrix
`ica`	$K\Omega^T K\Omega$, where Ω and **K** correspond to **W** and **K** definitions given in the fastICA R package
`msn`	$\Gamma\Lambda^2\Gamma^T$, where Γ is the matrix of canonical vectors corresponding to the X's found by canonical correlation analysis between X and Y, and Λ is the canonical correlation matrix
`msn2`	$\Gamma\Lambda\left(I-\Lambda^2\right)-1\Lambda\Gamma^T$
`gnn`	Θ, weights assigned to environmental data in canonical correspondence analysis
`randomForest`	No weight matrix

Number of neighbors k:

Default: k=1; Option to set k > 1

Option to use simple average of k (`mean`) or a distance weighted average (`dstWeighted`) of k

Search routines:

`ann = TRUE` → approximate nearest neighbor search = fast search method

Diagnostics:

Diagnostic statistics and plots are based on reference records. The following functions are available

`rmsd()` — computes root mean square difference (RMSD)

`compare()` — provides display of rmsd values for each of several imputation methods

`plot()` — matrix of plots of observed vs. imputed values

`errorStats()` — computes partitioning of error statistics as proposed by Stage and Crookston (2007)

Files

We will use tree data found in 'swodata.csv'. The data set contains information on 2462 Douglas-fir trees and 237 Pacific madrone trees collected in southwest Oregon. The variable names are defined in the R script. The R script used in this exercise can be found in 'yaImputeExample.r'.

Exercise

1. Extract only the Douglas-fir trees (`spp== "DF" & dead==0`) from the data set.
2. Randomly sample 20% of the new data set as target data and use the remaining 80% as reference data.
3. Define tree height (ht) and tree crown length (cl) as your variables of interest (Y-variables)
4. Define a set of tree- and stand-level variables as X-variables (e.g., "dbh","tph","bal","ba","ccfl")
5. Impute ht and cl for the target data set using MSN and randomForest imputation methods
6. Look at the diagnostic statistics (RMSD) available in the yaImpute package. **NOTE:** yaImpute only provides diagnostic statistics based on the reference data!
7. Since we know the actual values of ht and cl for the target data, look at RMSD calculated by the provided code

Some references on nearest neighbor imputation:

Crookston, N. L., and Finley. A.O. 2008. yaImpute: An R package for kNN imputation. Journal of Statistical Software 32(10):1-16.

Eskelson, B.N.I., Temesgen, H., LeMay, V., Barrett, T.M., Crookston, N.L., Hudak, A.T. 2009. The roles of nearest neighbor methods in imputing missing data in forest inventory and monitoring databases. Scandinavian Journal of Forest Research 24:235-246.

LeMay, V., and Temesgen, H. 2005. Comparison of nearest neighbor methods for estimating basal area and stems per hectare using aerial auxiliary variables. Forest Science 51:109-199.

Stage, A.R., and Crookston, N.L. 2007. Partitioning error components for accuracy-assessment of near-neighbor methods of imputation. Forest Science 53:62-72.