

VA: Software for Analyzing Verbal Autopsy Data¹

Gary King²

Ying Lu³

Version 0.9-2.9

June 16, 2008

¹Available from <http://GKing.Harvard.Edu/va> via a Creative Commons Attribution-Noncommercial-No Derivative Works 3.0, for academic use only

²David Florence Professor of Government, Harvard University (Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; <http://GKing.Harvard.Edu>, King@Harvard.Edu, (617) 495-2027).

³Assistant Professor, Department of Sociology, University of Colorado at Boulder (3E Ketchum Hall, UCB 327, Boulder, CO 80309. ylu@colorado.edu, 303-492-7030)

Contents

1	Introduction	2
2	Installation	2
2.1	Windows	2
2.2	Linux/Unix	2
3	Examples	3
3.1	Estimating Cause-specific Adult Mortality Fractions	3
3.2	Incorporating Additional Information Into Estimation	3
4	R Function Reference	4
4.1	Function <code>va()</code>	4
4.1.1	Usage	4
4.1.2	Inputs	4
4.1.3	Value	5
4.2	Function <code>va.gcv()</code>	5
4.2.1	Usage	6
4.2.2	Inputs	6
4.2.3	Value	7

1 Introduction

VA implements methods for estimating cause-specific mortality fractions based on verbal autopsy data introduced in King and Lu(2006):

Gary King and Ying Lu. 2007. "Verbal Autopsy Methods with Multiple Causes of Death," <http://GKing.Harvard.Edu/abs/vamc-abs.shtml>.

The package estimates cause-specific mortality rates in a population where a set of dichotomous symptoms are available, using the relationship between symptoms and a multicategory cause-of-death variable collected from a nearby medical facility. Estimation is nonparametric.

2 Installation

VA requires R version 2.2.0 or later, available from <http://cran.r-project.org/>. Installation of VA differs slightly by operating system.

2.1 Windows

Begin the installation process by launching R. At the R command prompt, type either:

```
> install.packages("VA", repos = "http://gking.harvard.edu")
```

to install using Gary King's website as the repository or

```
> install.packages("VA")
```

to install from the repository set as part of your options. Alternatively, you may download the Windows bundle from <http://GKing.Harvard.Edu/bin/windows/contrib> and use the R pull-down menu commands for installing a package from a zip file.

2.2 Linux/Unix

You initially need to create both local R and local R library directories if they do not already exist. At the Linux/Unix command prompt in your home directory, do this by typing:

```
> mkdir ~/.R ~/.R/library
```

Then open the '.Renviron' file that resides in your home directory, creating it if necessary, and add the line:

```
R_LIBS = "~/.R/library"
```

using your preferred text editor. These steps only need to be performed once. After starting R, install VA by typing at the R command prompt, either:

```
> install.packages("VA", repos = "http://gking.harvard.edu")
```

to install using Gary King's website as the repository or

```
> install.packages("VA")
```

to install from the repository set as part of your options. You can ignore warning messages. Alternatively, you may download the Linux/Unix bundle 'VA_XX.tar.gz', available from <http://gking.harvard.edu/R/CRAN/src/contrib/>, and place it in your home directory. Note that 'XX' is the current version number. Then, at the Linux/Unix command line from your home directory, type

```
> R CMD INSTALL VA_XX.tar.gz
```

to install the package.

3 Examples

3.1 Estimating Cause-specific Adult Mortality Fractions

In this section, we give an example of estimating cause-specific mortality fractions based on simulated data constructed from a verbal autopsy survey and related hospital deaths. In these verbal autopsy data, there are 764 registered deaths from hospital and 738 deaths collected from the general population. The total number of symptoms are 49, and there are 19 causes of reported deaths. Below we give the R syntax we use to analyze these data. The results are plotted in Figure 1.

```
> data(VAdata)
> FBA<-VAdata[VAdata$region==1,-1]
> DSS<-VAdata[VAdata$region==2,-1]
> res1<-va(formula=cbind(S1+...+S49)~cod,
  data=list(FBA,DSS), nsymp=16,
  n.subset=300,prob.wt=1,
  printit=TRUE,boot.se=FALSE)
```

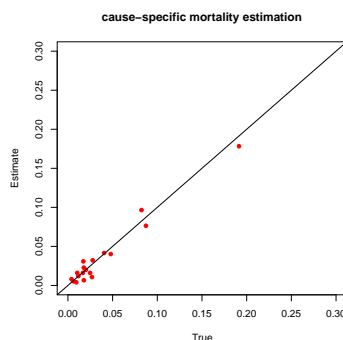


Figure 1: Estimation of Adult Cause-Specific Mortality Fractions. The observed cause-specific mortality is plotted horizontally and our verbal autopsy estimate is plotted vertically.

3.2 Incorporating Additional Information Into Estimation

Sometimes information about the rate of a specific cause of death is available from sources other than the verbal autopsy survey. For example, the death rate due to homicide and accidents can usually be determined by direct questioning with little error. The maternal death rate is typically lower than a certain known percentage due to sex and age group restrictions. In some other situations, a rough range of the death proportions for specific diseases may be available from other surveys on health outcomes. Researchers may also be willing sometimes to use analyses of neighboring countries or previous analyses in the same region to put bounds on some of the true proportions.

Below we give an example of how to incorporate such information into the `va` procedure. In this example, the cause of death “a13” and “a18” are fixed at 21.6% and 8%, and the fraction of cause “a14” is expected to be estimated between 0.5% and 1%.

```
res2 <- va(formula=cbind(S1+...+S49)~cod,
  fix=c("a13=0.216", "a18=0.08"),
  bound=c("0.005<a14<0.01"),
  data=list(FBA,DSS), nsymp=16,
  n.subset=10,prob.wt=1,printit=TRUE,boot.se=FALSE)
```

4 R Function Reference

4.1 Function `va()`

This function estimates the cause-specific mortality fractions from verbal autopsy data and hospital gold-standard cause-of-death diagnostics. It can also compute the bootstrap-based standard errors of the cause-specific mortality fractions.

4.1.1 Usage

```
va(formula, data=list(hospital, community), nsymp=16, n.subset=300,
method='quadOpt', fix=NA, bound=NA,
prob.wt=1, boot.se=FALSE, nboot=300, printit=TRUE, print.reg.size=FALSE)
```

4.1.2 Inputs

formula A formula object. The left side of the formula is the collection of symptoms. The right side is the cause of death. For example, if there are totally 5 symptoms, named `fever`, `coughing`, `chestpain`, `dizziness`, `shortbreath`, and the cause of death variable is `death`, then the formula can be written as:

```
formula=cbind(fever, coughing, chestpain, dizziness, shortbreath)~death
```

or for short:

```
formula=cbind(fever, ... ,shortbreath)~death
```

Note that the short way of writing formula requires the symptoms variables are located in a consecutive block in the data starting from `fever` and ending with `shortbreath`. Note that the current version requires the variable on the right hand side of the formula, `death` in this example, to be present in the `community` sample. If it is unknown in the `community` sample, the user needs to create such variable with arbitrary numerical values.

data A list of two datasets. The first is the hospital data, which contains a known cause of death for each individual, and a collection of symptoms from verbal autopsy studies. The second is the community data where typically only the symptoms are available from the verbal autopsy study. The known cause of death diagnostics may also be known in the community data if this is a validation study, but will not be used during estimation. Variable names must be exactly the same in two data sets.

nsymp A positive integer specifying the number of symptoms to be subset from all symptoms for estimating cause specific mortality fractions at each iteration. For the choice of `nsymp`, refer to King and Lu (2006). For practical purpose, we give the following recommendations: for total number of causes of death $D \leq 10$, use 7-12 symptoms; for $D > 10$, use 12-18 symptoms. If the number of observations is large in both hospital and community samples, for example, over 1000 cases total, use more symptoms, otherwise use fewer. Sensitivity analysis can also be used to choose `nsymp`. In general, the results stabilize in the right range of the choices of `nsymp`. Default=16.

n.subset A positive integer specifying the total number of draws of different subsets of symptoms. Default=300.

method A string specifying the computational procedure used to estimate the cause specific mortality fractions. When `method='quadOpt'`, CSMF is estimated via constrained quadratic programming. A subroutine (`Solve.QP`) from the `quadprog` package is called to perform the constrained quadratic optimization task. When `method='constrainLS'`, CSMF is estimated via constrained least squares. The default method is `quadprog` as it is faster and more stable.

- fix** A vector of strings that allow the user to fix a subset of the cause-specific mortality fractions to predetermined values chosen by the user (based on, e.g., the information obtained from other sources or prior knowledge). For example, setting `fix=c("malaria=0.15", "injuries=0.05")` fixes the mortality fractions due to malaria and injuries to 15% and 5%, respectively. Running `va` in this case will then attempt to allocate only the remaining 80% of the deaths. The default is `NA`, which means no constraint is imposed.
- bound** A vector of strings that allow the user to set fixed lower and upper bounds for a given subset of the cause specific mortality fractions (based on, e.g., information obtained from other sources or prior knowledge). For example, running `va` while setting `bound=c("0.2 < HIV < 0.35", "0.1 < TB < 0.15")` restricts the mortality fraction due to HIV to be between 20% and 35% and the TB rate is constrained to be between 10% and 15%. Causes not specified here are assumed to be bounded only by 0 and 1, and with the collection still constrained to the simplex. The default is `NA`, which means no constraint is imposed.
- prob.wt** A positive integer or a vector of weights that determines how likely a symptom is being selected in a subset. When `prob.wt` is a user input vector, it needs to be a vector of probabilities and sum up to 1. The length of `prob.wt` needs to be equal to the total number of symptoms. When `prob.wt=1`, binomial weights which are proportion to the inverse of variances of the each reported binary symptom variable. When `prob.wt=0`, all symptoms will be equally selected. Default=1.
- boot.se** A Logical value. If `TRUE`, bootstrap standard errors of the CSMF are estimated. This option typically takes a lot of computing time. The default is `FALSE`.
- nboot** A positive integer. If `boot.se=TRUE`, it specifies the number of bootstrapping samples taken to estimate the standard errors of CSMF. The default is 300.
- printit** Logical value. If `TRUE`, the progress of the estimation procedure is printed on the screen.
- clean.data** Logical value. If `TRUE`, `va` automatically deletes the symptoms variables(left-hand side of the formula) where there is no variation (all 0s or 1s). If `FALSE`, the user must make sure the data is cleaned before hand(which is recommended).
- print.reg.size** Logical value. If `TRUE`, the size of the regression matrix is printed at each step of sub-sampling. It provides helpful information for user to choose the number of symptoms to subsample. It is recommended to print the size of the regression matrix for different values of `nsymp` with a small size of `n.subset`.

4.1.3 Value

An object of class "VA", a list containing the following elements:

- est.CSMF** The estimated cause-specific mortality fractions.
- true.CSMF** The observed cause-specific mortality fractions, whenever available.
- est.se** The bootstrap standard errors of `est.CSMF` when `boot.se=TRUE`.
- true.CSMF.bootmean** The bootstrap mean of the observed cause-specific mortality fractions when they are available and when `boot.se=TRUE`.
- true.bootse** The bootstrap standard errors of the observed cause-specific mortality fractions when they are available and when `boot.se=TRUE`.

4.2 Function `va.gcv()`

This function use the method of general cross-validation to find the best `nsymp`, the number of symptoms to be subset from all symptoms for estimating the cause specific mortality fractions.

4.2.1 Usage

```
va.gcv(formula, data=list(hospital, community), nsymp.vec,  
        n.subset=300, prob.wt=1, boot.se=FALSE, nboot=1,  
        printit=FALSE, print.reg.size=TRUE)
```

4.2.2 Inputs

formula A formula object. The left side of the formula is the collection of symptoms. The right side is the cause of death. For example, if there are totally 5 symptoms, named `fever`, `coughing`, `chestpain`, `dizziness`, `shortbreath`, and the cause of death variable is `death`, then the formula can be written as:

```
formula=cbind(fever, coughing, chestpain, dizziness, shortbreath)~death
```

or for short:

```
formula=cbind(fever, ... ,shortbreath)~death
```

Note that the short way of writing formula requires the symptoms variables are located in a consecutive block in the data starting from `fever` and ending with `shortbreath`. Note that the current version requires the variable on the right hand side of the formula, `death` in this example, to be present in the `community` sample. If it is unknown in the `community` sample, the user needs to create such variable with arbitrary numerical values.

data A list of two datasets. The first is the hospital data, which contains a known cause of death for each individual, and a collection of symptoms from verbal autopsy studies. The second is the community data where typically only the symptoms are available from the verbal autopsy study. The known cause of death diagnostics may also be known in the community data if this is a validation study, but will not be used during estimation. Variable names must be exactly the same in two data sets.

nsymp.vec A vector of positive integer, containing different `nsymp` that can be used by `va()`. For a total of J number of causes of death and a total of `ns` symptoms in the sample, `nsymp.vec` can be set to be a vector `a:b`, while `a` is the smallest integer than $2^a > J$. `b` is typically set to be `floor(0.75*b)`. If sample size is small, `b` can be set to smaller value to avoid function exiting due to data sparsity. No default value is set.

n.subset A positive integer specifying the total number of subsets and thus estimations of all symptoms. The default is 300.

prob.wt A positive integer or a vector of weights that determines how likely a symptom is of being selected for a subset. When `prob.wt` is a user input vector, it needs to be a vector of probabilities and sum up to 1. The length of `[prob.wt]` needs to be equal to the total number of symptoms. When `prob.wt=1`, binomial weights which are proportion to the inverse of variances of the each reported binary symptom variable. When `prob.wt=0`, all symptoms will be equally selected. The default is 1.

boot.se A Logical value. If `TRUE`, bootstrap standard errors of the CSMF are estimated. This typically takes a lot of computing time. It is highly suggested to set `boot.se=FALSE` in `va.gcv`. Default=`FALSE`.

nboot A positive integer. If `boot.se=TRUE`, it specifies the number of bootstrapping samples taken to estimate the standard errors of CSMF. The default is 1.

printit Logical value. If `TRUE`, the progress of the estimation procedure will be printed on the screen.

print.reg.size Logical value. If **TRUE**, the size of the regression matrix is printed at each step of subsampling. It provides helpful information for user to choose the number of symptoms to subsample. It is recommended to print the size of the regression matrix for different values of **nsymp** with a small size of **n.subset**.

4.2.3 Value

An object of class “VA”, a list containing the following elements:

best.symp It returns the **nsymp**, the size of subset that minimizes mean square error between estimated cause-specific mortality fraction and the observed cause-specific mortality fraction.

mse It is a vector of mean square errors associated with each size of the subsets (as specified in **symp.vec**).

References