Package ‘ssd’

February 20, 2015

Type Package

Title Sample Size Determination (SSD) for Unordered Categorical Data

Version 0.3

Date 2014-11-30

Author Junheng Ma and Jiayang Sun

Maintainer Junheng Ma <jxm216@case.edu>

Description ssd calculates the sample size needed to detect the differences between two sets of unordered categorical data.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-01 09:28:44

R topics documented:

`ssd` *Sample Size Determination (SSD) for Unordered Categorical Data*

Description

ssd calculates the required sample sizes n1 and n2 needed to detect the differences between two populations of unordered k-cell categorical data, using seven methods. The basic test statistic is the chi-square statistic: sum[nij-Ej]^2/Ej, where the summation is over i=1,2, and j=1,..,k; nij is the observed cell frequency for cell j, of population i, and Ej is the estimate of expected frequency for cell j under the null hypothesis H0: p1=p2. Here p1=(p11,..,p1k) and p2=(p21,..,p2k) are the k-dimensional true probability vectors for populations 1 and 2. Depending on the specification of input parameters, sample sizes (n,n) or simply n in the balanced case or (n1,n2) in the unbalanced case are provided under two schemes by up to seven methods: the “original” method and its 6 improvements, “minimum difference”, “correction”, “bootstrap mean”, “bootstrap median”, “bootstrap 75th percentile”, and “bootstrap 80th percentile” methods.

Usage

```
ssd(p1, p2, k, ratio = 1, alpha = 0.05, beta = 0.2, cc = 0.02,
    d = 0.2, r = 0.3, m, scheme = 'M2', Niter = 500)
```

Arguments

`p1,p2` Either specified, or estimated probability vectors from pilot data for the first and second groups of categorical data. Required parameter for scheme M2.

`k` Dimension of the categorical data. Required parameter for scheme M1.

`ratio` Ratio (=n1/n2) between pilot sample sizes of two groups of categorical data. It is a required parameter for unbalanced case. Default value is 1, the balanced case.

`alpha` Significance level of the test. Default value is 0.05.

`beta` False negative rate of the test. Default value is 0.2, i.e., power=0.8.

`cc` Minimum difference needed to detect between two probability vectors: “|p1j-p2j|=>cc” for all j. Here “=>” means “greater than or equal to.” Required parameter for “minimum difference” method. Default value is 0.02. Optional parameter for scheme M2.

`d` Specified minimum average difference between p1 and p2. The d is such that “ave_j|p1j-p2j|=>d” and represents a difference that a scientist cares to detect. Default value of d is 0.2. Required parameter for scheme M1.

`r` Specified minimum relative difference between p1 and p2. The r is such that “min_j|p1j-p2j|/pj=>r” and represents a relative difference that one cares to detect, where pj=(p1j+p2j)/2. Default value is 0.3. Required parameter for scheme M1.

`m` Pilot sample size of the first group of categorical data, required for “correction” and bootstrap methods. Optional parameter for scheme M2.

`scheme` Scheme used to calculate the sample sizes. Possible values are ’M1’ and ’M2’. Default value is ’M2’. ’M1’ requires (k,d,r) where d and r are specified AVERAGE and RELATIVE minimal differences between p1 and p2 that a scientist cares to detect. ’M2’ requires hypothetical p1 and p2, or computed p1 and p2 from pilot data.

`Niter` Number of iterations used by the “correction” and bootstrap methods. Default value is 500. Optional parameter for scheme M2.

Details

Under ’M1’, k, d, r are required. The default values are d=0.2 and r=0.3. The M1 currently only allows for the balanced case. Under ’M2’, values of p1 and p2 are required. Without pilot data, they are chosen so that their differences roughly represent the perceived, or the smallest differences that an investigator cares to detect. With pilot data, p1 and p2 are the estimates from the pilot data.

For example, p1=(0.10,0.25,0.30,0.20,0.15) and p2=(0.15,0.20,0.25,0.30,0.10) represent small difference, p1=(0.10,0.25,0.30,0.20,0.15) and p2=(0.17,0.32,0.36,0.10,0.05) represent medium difference, p1=(0.10,0.25,0.30,0.20,0.15) and p2=(0.30,0.10,0.20,0.10,0.30) represent large difference. Under M2, in addition to p1 and p2, cc is required for "minimum difference" method, and (n1,Niter) are required parameters for "correction" method and 4 bootstrap methods. They should be specified if a user wants to take values different from the default values.

Value

Depending on the parameter specification, sample sizes are provided by up to 7 methods.

For balanced case:

`n`	Calculated common sample size for two populations

For unbalanced case:

`n1,n2`	Calculated sample sizes for first and second populations

Note

Practical Advice 1: When comparing k categories of two multinomial populations, if some categories are known to have same or similar counts for the two populations, it is highly recommended to remove these categories to reduce to comparing k'(<k) categories. This is not only practical, but also generally results in a smaller n needed to compare these k' categories than n computed based on the original k categories. This was called scheme 'M3' method in Ma, Sun, and Sedransk (2014).

Practical Advice 2: How to choose (n1,n2) among 7 methods:

1. If neither p1 and p2 nor pilot data are available, use scheme M1. If the hypothetical p1 and p2 are given, the "original" method under scheme M2 will provide the BASIC calculated sample size(s).

2. If a pilot sample of size m>=10 is available, this package will output the sample sizes computed from the 7 methods under scheme M2 (the default). If you are confident with your specified p1 and p2, use the sample size computed by the "original" method (the first output). If p1 and p2 are estimated values that one is not sure and the resulting sample sizes are large, use the following combined approach for choosing one of the 7 results:

2.1. If a reliable minimum difference, cc, is available, the "minimum difference" method should be used. A default result for cc=0.02 is provided.

2.2. If cc is not available or you do not like the default value, use the output from one of the other 6 methods. If all the estimates are large, remove some small categories as suggested in Practical Advice 1 or conduct an additional pilot study.

2.3. For small and moderate pilot sample sizes, use the estimate from the "bootstrap 80th percentile" method. For a fairly large pilot sample size, use the estimate from the "bootstrap 75th percentile" method. For a very large pilot size, use the estimate from the "bootstrap mean" method.

Practical Advice 3: If one does not know how large an n is too large, he/she can compute n using scheme M1 with reasonable r and d. This n can then be considered as a cap of the computed sample size.

Author(s)

Junheng Ma and Jiayang Sun

References

Ma, J., Sun, J. and Sedransk, J. (2014) Modern Sample Size Determinations for Unordered Categorical Data. Statistics and Its Interface, **7**, 2, 219–233.

Examples

```
# Scheme M1 for Balanced case
# Case1: k=5, using default d=0.2, and r=0.3 under M1
ssd(k = 5, scheme = 'M1')
# Case2: k=5, specify d and r as below under M1
ssd(k = 5, d = 0.3, r = 0.35, scheme = 'M1')
ssd(k = 5, d = 0.1, r = 0.2, scheme = 'M1')

# Scheme M2 for Balanced case
# If p1 and p2 are true parameters, the original method provides
# the exact sample size needed:
p1 <- c(0.10, 0.25, 0.30, 0.20, 0.15)
p2 <- c(0.17, 0.32, 0.36, 0.10, 0.05)
ssd(p1, p2)
# If p1 and p2 are estimates from pilot data, the output from
# the "minimum difference" method will be also given if
# the minimal difference cc is specified, while the output from
# "correction" and 4 bootstrap methods are provided if m,
# the pilot sample size is also specified. One can change the
# default value of Niter=500
ssd(p1, p2, m = 100)
ssd(p1, p2, m = 40)
# Non-bootstrap estimates provide similar answers in this case.
# The bootstrap median underestimates as it usually does, and
# bootstrap 75th is better than 80th in this case for m=100.
# With smaller m, bootstrap 80th method is better if p1 and p2
# were true parameters.

# Scheme M2 for Unbalanced case
ssd(p1, p2, m = 100, ratio = 0.67)
# Bootstrap 75th method is better than 80th method for m=100.
```

Index