UStat Package Manual

July 16, 2025

1 Overview

This document provides some more detailed documentation for the functions to compute the $U-$statistic estimators of the variance-covariance and its sampling covariance proposed by Rose, Schellenberg, and Shem-Tov (2022). This package is available on PyPI here (link forthcoming).

2 Functions

2.1 varcovar

The 'ustat.varcovar(A,C)' function computes the unbiased covariance between two datasets A and C. The function also supports weighted variance calculations (where each weight corresponds to a row of A and C) and weighting by year. Specifically, the function calculates any of the following:

(1) Unweighted:

$$\hat{C}_{unweighted} = \left(\frac{J-1}{J}\right)\frac{1}{J}\sum_{j=1}^{J}\binom{T_j}{2}^{-1}\sum_{t=1}^{T_j-1}\sum_{k=t+1}^{T_j}\bar{Y}_{jt}^{A}\bar{Y}_{jk}^{C} - \frac{2}{J^2}\sum_{j=1}^{J-1}\sum_{k>j}^{J}\bar{Y}_j^{A}\bar{Y}_k^{C} \tag{1}$$

(2) Weighting each individual

$$\hat{C}_{w} = \sum_{j=1}^{J}\binom{T_j}{2}^{-1}\tilde{w}_j(1-\tilde{w}_j)\sum_{t=1}^{T_j-1}\sum_{k=t+1}^{T_j}\bar{Y}_{jt}^{A}\bar{Y}_{jk}^{C} - 2\sum_{j=1}^{J-1}\sum_{k>j}^{J}\tilde{w}_j\bar{Y}_j^{A}\tilde{w}_k\bar{Y}_k^{C} \tag{2}$$

(2) Weighting each individual by years observed

$$\hat{C}_{w} = \sum_{j=1}^{J}\frac{\tilde{T}_j^{A\wedge C} - \tilde{T}_j^{A}\tilde{T}_j^{C}}{|T_j^{A\wedge C}|(|T_j^{A\wedge C}|-1)}\sum_{t\in T_j^{A\wedge C}}\sum_{k\neq t}^{k\in T_j^{A\wedge C}}\bar{Y}_{jt}^{A}\bar{Y}_{jk}^{C} - 2\sum_{j=1}^{J-1}\sum_{k>j}^{J}\tilde{w}_j\bar{Y}_j^{A}\tilde{w}_k\bar{Y}_k^{C} \tag{3}$$

where $\tilde{w}_j = w_j/\sum_{j=1}^{J}w_j$, $\tilde{T}_j^A = |T_j^A|\sum_{j=1}|T_j^A|$, and $|T_j^A|$ represents the number of time periods individual j is observed for outcome A.

Note, this function can yield negative variance estimates due to the debiasing procedure. Negative variance estimates occur when the variance of teacher means is close to 0.

2.1.1 Arguments

ustat.varcovar(A,C, w, yearWeighted=False, quiet=True)

1. A, C = two J-by-T arrays between which you want to calculate the variance-covariance. A, C can contain missing values (in the form of a Nan), and each row of A and C can have missings in different spots.

2. w = an array of length J containing weights for the rows of A, C. Used to compute a weighted variance-covariance.

3. yearWeighted = option to compute weights based on the number of time periods each row is observed. Supports missing values in the same way as A, C.

4. quiet = whether to report to user what type of variance was calculated and whether the panels were balanaced/unbalanced. Reporting messages suppressed by default.

2.1.2 Usage

```
import ustat as ustat
import numpy as np

# Data and weights
np.random.seed(48912)
n_teachers, n_time = 50, 10
X, Y = ustat.generate_unique_nan_arrays(n_rows=n_teachers, n_cols=n_time, n_arrays=2,
    min_int=1, max_int=9, nan_prob=0.25, seed = 48912, balanced = False)

weights = np.random.exponential(size = n_teachers)

# Variance-covariance
ustat.varcovar(X, X) # Var(X)
ustat.varcovar(X, Y) # Cov(X, Y)

ustat.varcovar(X, X, w = weights) # weighted Var(X)
ustat.varcovar(X, Y, w = weights) # weighted Cov(X, Y)

ustat.varcovar(X, X, yearWeighted = True) # year weighted Var(X)
ustat.varcovar(X, Y, yearWeighted = True) # year weighted Cov(X, Y)
```

2.2 ustat_samp_covar

The 'ustat.ustat_samp_covar(A,B, C, D)' function computes the sampling covariance of $Cov(A, B)$ and $Cov(C, D)$. Note that we do not impose any logical cap on the sampling variance, meaning this function can yield sampling covariances-variances which imply correlations exceeding 1. Specifically, the function computes an estimator for:

$$
\begin{aligned}
&Cov\Big(\hat{Cov}(a_j^A, a_j^B) - Cov(a_j^A, a_j^B), \hat{Cov}(a_j^C, a_j^D) - Cov(a_j^C, a_j^D)\Big) = \\
&\quad \sum_i \sigma_i^{AC}\Big(\sum_{k\neq i} C_{ik}^{AB} a_{j(k)}^B\Big)\Big(\sum_{k\neq i} C_{ik}^{CD} a_{j(k)}^D\Big) + \sum_i \sigma_i^{AD}\Big(\sum_{k\neq i} C_{ik}^{AB} a_{j(k)}^B\Big)\Big(\sum_{k\neq i} C_{ik}^{DC} a_{j(k)}^C\Big) \\
&\quad + \sum_i \sigma_i^{BC}\Big(\sum_{k\neq i} C_{ik}^{BA} a_{j(k)}^A\Big)\Big(\sum_{k\neq i} C_{ik}^{CD} a_{j(k)}^D\Big) + \sum_i \sigma_i^{BD}\Big(\sum_{k\neq i} C_{ik}^{BA} a_{j(k)}^A\Big)\Big(\sum_{k\neq i} C_{ik}^{DC} a_{j(k)}^C\Big) + \\
&\qquad \sum_i \sigma_i^{AD} \sum_{k\neq i} C_{ik}^{AB} C_{ik}^{DC} \sigma_k^{BC} + \sum_i \sigma_i^{AC} \sum_{k\neq i} C_{ik}^{AB} C_{ik}^{CD} \sigma_k^{BD} \qquad (4)
\end{aligned}
$$

where σ_i^{AC} represents the covariance between A and C and

$$
C_{ik}^{AC} = \begin{cases} \frac{J-1}{J^2} \frac{1}{|T_j^A||T_j^C| - |T_j^A \cap T_j^C|} & \text{if } j(i) = j(k) \\ \frac{-1}{|T_{j(i)}^A||T_{j(k)}^C| J^2} & \text{if } j(i) \neq j(k) \end{cases}
$$

Note, the code computes *unbiased* estimators of the product-sums $\Big(\sum_{k\neq i} C_{ik}^{AB} a_{j(k)}^B\Big)\Big(\sum_{k\neq i} C_{ik}^{CD} a_{j(k)}^D\Big)$. As with the variance-covariance estimator embodied in varcovar(), this means estimated sampling variances *can* be negative, though this does not happen often.

2.2.1 Arguments

ustat.ustat_samp_covar(A,B, C, D)

1. A, B, C, D = four J-by-T arrays. Each can contain missing values (in the form of a Nan), and each row of each array can contain missing values in different spots.

2.2.2 Usage

```
import ustat as ustat
import numpy as np

# Data and weights
np.random.seed(48912)
n_teachers, n_time = 50, 10
A, B, C, D = ustat.generate_unique_nan_arrays(n_rows=n_teachers, n_cols=n_time, n_arrays=4,
    min_int=1, max_int=9, nan_prob=0.25, seed = 48912, balanced = False)

# Compute
ustat.ustat_samp_covar(A, A, A, A) # Var(Var(A))
ustat.ustat_samp_covar(A, B, A, B) # Var(Cov(A, B))
ustat.ustat_samp_covar(A, B, C, D) # Cov(Cov(A, B), Cov(C, D))
```