Computational Statistics (2024) 39:2899–2924
https://doi.org/10.1007/s00180-023-01408-7

ORIGINAL PAPER

A general stream sampling design

Bardia Panahbehagh[1,2] · Raphaël Jauslin[1] · Yves Tillé[2]

Received: 25 May 2023 / Accepted: 21 August 2023 / Published online: 20 September 2023

Abstract
With the emergence of the big data era, the need for sampling methods that select samples based on the order of the observed units is felt more than ever. In order to meet this necessity, a new sequential unequal probability sampling method is proposed. The decision to select or not each unit is made based on the order in which the units appear. A variant of this method allows a selection of a sample from a stream. This method consists in using sliding windows which are a kind of strata of controllable size. This method also allows the sample to be spread in a controlled manner throughout the population. A special case of the method with windows of size one leads to deciding on each sampling unit immediately after observing it. The implementation of size one windows is simple and will be presented here based on an algorithm with a single condition. Also, by selecting the windows of size two, we will have one of the optimal stream sampling methods, which results in a well-spread stream sample with positive second-order inclusion probabilities.

Keywords Stream sampling algorithms · Inclusion probability · Flow sampling · Stratified stream sampling · Window

1 Introduction

There are dozens of methods of unequal probability sampling. Most of them are described in the books of Hanif and Brewer (1980), Gabler (1990), Tillé (2006) and Chaudhuri and Pal (2022). Nevertheless, very few methods allow you to decide to select the units in the order in which they appear. Systematic sampling with unequal probabilities (Madow 1949) allows sequential sampling from a stream. However, once the first units are examined, this method quickly becomes deterministic, making it predictable, which can be problematic. Indeed, sampling should not be predictable if it is used to perform a control (Busnel and Tillé 2020).

✉ Bardia Panahbehagh
panahbehagh@khu.ac.ir

[1] Department of Mathematics, Kharazmi University, Tehran, Iran

[2] Institute of Statistics, University of Neuchâtel, Neuchâtel, Switzerland

Several methods are already available for sampling in a stream. The sequential version of Hanurav-Vijayan sampling design (Vijayan 1968; Aubry 2023) can be considered as a generalization of the Sunter sampling procedure (Sunter 1977, 1986). The ordered pivotal method (as a special case of splitting method of Deville and Tillé 1998) and Deville's systematic sampling are two different implementations of the same sampling design (Chauvet 2012). A disadvantage of the order pivotal method is that the decision to take or not the first unit may be made after examining a very large number of units in the sampling frame. The sequential version of Hanurav-Vijayan is not generally consistent using Narain-Horvitz-Thompson (Narain 1951; Horvitz and Thompson 1952) estimator. The second-phase of this design may be more simply implemented in terms of a sequential procedure (Chauvet 2022), and then can not respect the order of the observations completely. Disadvantages of Deville's systematic method include numerous calculations for each step and problems with receiving data in groups in the middle of the process. In addition to these algorithms, reservoir sampling methods are available to collect samples from a stream (Chao 1982; Cohen et al. 2009; Tillé 2019). Using the idea of reservoir in data mining approaches, such as pattern sampling (Boley et al. 2011; Diop et al. 2018), has provided powerful tools for mining online stream populations to discover patterns (Giacometti and Soulet 2021). In such methods, the reservoir is updated each time a unit enters. The final decision on certain units will then be made very late. In stream sampling, it is a convenient property to be able to comment on each unit immediately after observing it.

Chromy sequential method (Chromy 1979) leads to an immediate sampling algorithm with the same design of Deville's systematic and order pivotal methods (Chauvet 2021). Many zero second-order inclusion probabilities are the main drawback of the three methods. To solve this problem, (Chromy 1979) has proposed to partially randomize the order of the units in the population before applying the sampling algorithm. With such a reorganization of population units, we lose the applicability of stream methods. Here, to overcome all the discussed drawbacks, we first propose a method that allows us to decide on the selection of units according to the order of their appearance in a list. The probabilities are updated only in a window containing a very small number of units after the unit for which the decision has been made. The method is therefore particularly interesting for selecting units from a stream. The great advantage of this new method is that it forces the decision on the units according to their order of appearance in the stream.

Recently, Jauslin et al. (2022) presented a two-phase balanced version of sequential sampling that respects the order of units in the sample selection process. But, depending on the auxiliary variables, this method may not lead directly to a sample. Therefore, the decision regarding some units will be postponed until the completion of the first phase of sampling, and during the implementation of the landing phase of balanced sampling.

We then extend the method to apply integer-sized windows where the size of each window is defined as the expected number of sample units within it. With windows of size two, simultaneously we take care about three important issues in streaming populations, the positivity of second-order inclusion probabilities, applicability on streaming populations and finally spreading the sample along the population indexes. In this version of the method, immediately after observing enough units with inclusion probabilities sum to greater than or equal to two, we can decide for all of them and two of them will be selected.

Also, as a special version of the method, windows of size one is equivalent to Chromy method. It can also easily be shown that Deville's systematic and Chromy method lead to the same sampling design. The method is therefore quite appropriate for stream data and can be applied to unequal or equal inclusion probabilities.

To discuss these matters, in Sect. 2, we introduce the notations and the principal concept of sampling theory. In Sect. 3, we present the proposed sampling method while, in Sect. 4, we extend the method to stream sampling. Section 5 is devoted to the method for windows of size integers and calculation of the second-order inclusion probabilities. In Sect. 6, we present a stream sampling with windows of size one with extending it to an immediate decision sampling. In Sect. 7, we discuss the size of the windows. The calculation of the second-order inclusion probabilities of the method is presented in Sect. 5. In Sect. 8, we run some small examples and simulations. The manuscript ends with a conclusion on the proposed methods, in Sect. 9.

2 Notation

Consider a finite population of size N labeled by $U = \{1, 2, \dots, N\}$, a vector of values taken by a variable of interest $\boldsymbol{y} = (y_1, y_2, \dots, y_N)^\top$ and a vector of first-order inclusion probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)^\top$, each associated with the respective labels in U. Consider also the total of the variable of interest,

$$Y = \sum_{k \in U} y_k,$$

as the main parameter. A sample s is a subset of U and a sampling design $p(.)$ is a probability distribution on all the subsets of U. A random sample is defined by:

$$\Pr\,(S = s) = p(s),\; p(s) \geq 0,\; \text{and} \sum_{s \subset U} p(s) = 1,\; \text{for all}\; s \subset U.$$

To estimate this parameter, two approaches can be used:

(1) Consider a sampling design $p(.)$ and find a sampling method (or algorithm) to implement this design. Based on this approach, according to $p(.)$, all the first- and second-order inclusion probabilities can be specified using

$$\pi_k = \Pr(k \in S) = \sum_{s \ni k} p(s) \text{ and}$$

$$\pi_{k\ell} = \Pr(k, \ell \in S) = \sum_{s \ni \{k,\ell\}} p(s),\; \text{for all}\; k, \ell \in U.$$

(2) Consider a vector of first-order inclusion probabilities $\boldsymbol{\pi}$ and find a sampling method to select a random sample S such that $\Pr(k \in S) = \pi_k$, for all $k \in U$.

In this article, the second approach is considered. If $\pi_k > 0$, for all $k \in U$, then Y can be estimated unbiasedly employing the first-order inclusion probabilities using the Narain-Horvitz-Thompson estimator

$$\widehat{Y} = \sum_{k \in S} \frac{y_k}{\pi_k}.$$

Its accuracy depends on the second-order inclusion probabilities

$$\operatorname{Var}(\widehat{Y}) = \sum_{k \in U} \sum_{\ell \in U} (\pi_{k\ell} - \pi_k \pi_\ell) \frac{y_k}{\pi_k} \frac{y_\ell}{\pi_\ell}. \tag{1}$$

Indeed, the values of $\boldsymbol{\pi}$ indicate the desired chance (calculated using some auxiliary variables) for the units to be selected in the final sample and different sampling methods can lead to the same $\boldsymbol{\pi}$. Methods with the same $\boldsymbol{\pi}$, can lead to different matrices of second-order inclusion probabilities $\boldsymbol{\Pi}$, which directly affects the accuracy of the estimation. If $\pi_{k\ell} > 0$, for all $k, \ell \in U$, then $\operatorname{Var}(\widehat{Y})$ can be unbiasedly estimated by

$$\widehat{\operatorname{Var}}(\widehat{Y}) = \sum_{k \in S} \sum_{\ell \in S} \frac{\pi_{k\ell} - \pi_k \pi_\ell}{\pi_{k\ell}} \frac{y_k}{\pi_k} \frac{y_\ell}{\pi_\ell}.$$

The accuracy is however not always the most important criterion for evaluating a strategy (combination of a sampling method and an estimator). Another important criterion can be the applicability of the strategy according to the accessibility of the data over time. Since efficient designs are usually complex, it is important to be able to calculate the second-order inclusion probabilities to have a vision of the accuracy of designs. These topics will be discussed further in Sect. 4.

3 One-step one-decision sampling method

In this section, we present a procedure called One-Step One-Decision (OSOD) sampling method. The general idea of the method is that, at each step, a decision is made about the selection of the unit. After that, inclusion probabilities are updated according to the decision made on this unit. This procedure is repeated on the following units and it takes at most N steps to obtain the final sample. This method does not allow the selection of a sample from a stream but serves as a basis for the construction of the following methods.

Consider a finite population U. Define the sum of the inclusion probabilities

$$n = \sum_{k \in U} \pi_k,$$

which can be non-integer. For simplicity, we first consider the method in which the decision is made for the first unit. Also suppose that

$$\sum_{k=1}^{N} \pi_k \geq 1 \text{ and } \sum_{k=1}^{N}(1-\pi_k) \geq 1.$$

The first unit is selected with probability π_1. Let π_1^1 be the updated inclusion probability of unit 1. It can be 0 or 1 and can be summarized as follows

$$\pi_1^1 = \begin{cases} 1 & \text{with probability } \pi_1 \\ 0 & \text{with probability } 1-\pi_1. \end{cases} \tag{2}$$

After deciding on the first unit, the remaining inclusion probabilities are updated. The general idea of the method is to increase (respectively decrease) the inclusion probabilities of the following units depending on whether π_1^1 has been changed to 0 (respectively 1). The inclusion probabilities of the remaining units are updated as follows

$$\pi_k^1 = \begin{cases} \pi_k^{1(0)} = \min(c_1\pi_k, 1) & \text{if } \pi_1^1 = 0 \\ \pi_k^{1(1)} = \dfrac{\pi_k - \pi_k^{1(0)}(1-\pi_1)}{\pi_1} & \text{if } \pi_1^1 = 1, \end{cases} \quad \text{for } k = 2, \dots, N, \tag{3}$$

where constant c_1 is defined by

$$\sum_{k=2}^{N} \min(c_1\pi_k, 1) = n. \tag{4}$$

Equation (3) is calculated in such a way that the overall chances of the units to be selected are exactly equal to the predetermined inclusion probabilities, which results in respecting the sample size expectation. Indeed, in the first part of (3), the first unit is not selected, and then we have distributed π_1 on the other units as $\pi_k^{1(0)} = \pi_k + \alpha_k\pi_1$, for some $0 < \alpha_k \leq 1$, using (4) which guaranties that α_k is proportional to π_k, and the updated inclusion probabilities do not exceed the value of 1. The second part of (3) has been built in a way to preserve the first order inclusion probabilities.

After the first step, the decision is irrevocably made for unit 1. Simply, this operation can be repeated for other steps $t = 2, \dots, N$. At step t, decisions are made on the first $t-1$ units and $\boldsymbol{\pi}$ has been updated $t-1$ times. The updated vector is denoted by

$$\boldsymbol{\pi}^{t-1} = (\pi_1^{t-1}, \pi_2^{t-1}, \dots, \pi_k^{t-1}, \dots, \pi_N^{t-1})^\top,$$

where its first $t-1$ units are in $\{0, 1\}$. Again, it is supposed that

$$\sum_{k=t}^{N} \pi_k^{t-1} \geq 1 \text{ and } \sum_{k=t}^{N}(1-\pi_k^{t-1}) \geq 1, \tag{5}$$

and a decision is taken for the t^{th} unit with probability

$$\pi_t^t = \begin{cases} 1 & \text{with probability } \pi_t^{t-1} \\ 0 & \text{with probability } 1 - \pi_t^{t-1}. \end{cases}$$

Then, the inclusion probabilities are updated by

$$\pi_k^t = \begin{cases} \pi_k^{t(0)} = \min(c_t \pi_k^{t-1}, 1) & \text{if } \pi_t^t = 0 \\ \pi_k^{t(1)} = \dfrac{\pi_k^{t-1} - \pi_k^{t(0)}(1 - \pi_t^{t-1})}{\pi_t^{t-1}} & \text{if } \pi_t^t = 1, \end{cases} \quad \text{for } k = t+1, t+2 \ldots, N,$$

where c_t is defined by

$$\sum_{k=t+1}^{N} \min(c_t \pi_k^{t-1}, 1) = n - n_t,$$

and n_t is the number of selected units up to step t.

Conditions (5) are required to update other probabilities. For example, at the first step, it is necessary to have enough amplitude in other inclusion probabilities to distribute π_1 as $\pi_k^{1(0)} = \pi_k + \alpha_k \pi_1$ (if unit 1 is not selected) or to remove $1 - \pi_1$ from them as $\pi_k^{1(1)} = \pi_k - \alpha_k(1 - \pi_1)$ (if unit 1 is selected) for some $0 < \alpha_k \leq 1$ on the following inclusion probabilities. Then, henceforth, OSOD stands for the design with a population that satisfies Conditions (5).

The procedure is formally presented in Algorithm 1.

Algorithm 1 OSOD

```
Initialize with $n_0 = 0$, $\boldsymbol{\pi}^0 = \boldsymbol{\pi}$ and $n = \sum_{k \in U} \pi_k$.
for $t = 1, 2, 3, \ldots, N$ do
    Calculate $c_t$ such that $\sum_{k=t+1}^{N} \min(c_t \pi_k^{t-1}, 1) = n - n_{t-1}$ where $n_{t-1} = \sum_{k=1}^{t-1} \pi_k^{t-1}$,
    Generate $u$, a realization of a uniform random variable in $[0, 1]$,
    if $u > \pi_t^{t-1}$ then
        Set $\pi_t^t = 0$ and $\pi_k^t = \min(c_t \pi_k^{t-1}, 1)$ for all $k > t$,
    else
        Set $\pi_t^t = 1$ and $\pi_k^t = \frac{\pi_k^{t-1} - \min(c_t \pi_k^{t-1}, 1)(1 - \pi_t^{t-1})}{\pi_t^{t-1}}$ for all $k > t$,
    end if
    Set $\pi_k^t = \pi_k^{t-1}$, for all $k < t$.
end for
```

Result 1 shows that the sampling procedure respects the inclusion probabilities and thus the fixed sample size.

Result 1 With OSOD, for all $t = 1, 2, \ldots$ we have

$$\mathrm{E}(\pi_k^t) = \pi_k, \quad \text{for all } k \in U, \qquad \text{and} \qquad \sum_{k=1}^{N} \pi_k^t = \sum_{k=1}^{N} \pi_k.$$

The proof of Result 1 and all the following results are given in Appendix.

Currently, the sampling procedure considers the entire population to modify the inclusion probabilities. In the next section, an improvement is proposed for sampling from a stream. In fact, only a small part of the following units is enough to update the inclusion probabilities. This improvement, completely modifies the application of the method.

4 Extending the method to stream sampling

A data stream is a sequence of information arriving sequentially in time. When sampling streams, there is usually a very large set of data coming in continuously. Due to the limited space required for data storage, it is desirable to make a decision on the selection of units at the right time.

As pointed out in Sect. 1, in the methods proposed so far, the decision about a unit that is at the beginning of a stream can be made too late. Ideally, the decision to select or not a unit from a stream must be taken as soon as possible, which the OSOD method allows. The idea to extend the OSOD method to stream sampling consists of considering a window of units instead of the entire population to update inclusion probabilities.

Result 2 shows that the number of units required to be considered, depends on the inclusion probabilities of the following units. Essentially, it is enough to wait until we have a set of units such that there exists a constant c_t that satisfies $\pi_t \geq (1 - 1/c_t)$. Furthermore, Result 3 also shows that a sufficient condition to update inclusion probabilities is that the inclusion probabilities within the window sum up to an integer.

Result 2 In OSOD, at step t, a necessary and sufficient condition to have enough space to update the following inclusion probabilities is that

$$\pi_t \geq 1 - 1/c_t.$$

Result 3 In OSOD, at step t, a sufficient condition to have enough space to update the following inclusion probabilities is that $n = \sum_{k \in U} \pi_k$ is an integer.

Therefore, after deciding on the first unit, we need a sufficient number of units with c_1 that satisfies $\pi_1 \geq (1 - 1/c_1)$. Moreover, if there is a window of integer size, then it is possible to apply the method on this window. Still, it could be possible that no window sums up to an integer and no constant c_1 satisfies the condition until the end of the stream. In this case, it is possible to use Result 3 and the notion of phantom unit (Grafström et al. 2012) to artificially transform the sum of the inclusion probabilities to an integer and finish the procedure. Also, an interesting method to solve this problem will be presented in Sect. 5.

Algorithm 2 gives a detailed explanation of the proposed method.

Algorithm 2 OSOD Stream Sampling

```
Initialize with $\pi_1^0 = \pi_1$,
for $t = 1, 2, 3, \ldots$ do
    if there exist a window of size $m$, in which $\pi_t^{t-1} \geq (1 - 1/c_t)$, where
        $$\sum_{k=t+1}^{t+m} \min(c_t \pi_k^{t-1}, 1) = \sum_{k=t}^{t+m} \pi_k^{t-1}$$
      then
        1) Set $n - n_t = \sum_{k=t}^{t+m-1} \pi_k^{t-1}$, $N = t + m - 1$,
        2) Implement Algorithm 1, just for one step at $t$.
    else
        1) Consider the largest possible window (all the available data), say of size $M$,
        2) Add a phantom unit $k = t + M$ with
            $$\pi_{t+M}^{t-1} = 1 - \left( \sum_{k=t}^{t+M-1} \pi_k^{t-1} - \left\lfloor \sum_{k=t}^{t+M-1} \pi_k^{t-1} \right\rfloor \right),$$
        3) Set $n_t = 0$, $n = \sum_{k=t}^{t+M} \pi_k^{t-1}$ and $N = t + M$,
        4) Implement Algorithm 1 from step $t$ for $M$ steps,
        5) Delete the phantom unit with $\pi_{t+M}^{t+M-1}$,
        6) Set $t = t + M$.
    end if
end for
```

In summary, OSOD has many advantages on streams:

- The decision about units can be made based on their order.
- In order to update the population inclusion probabilities after selecting a unit, the entire population is not needed and only a small part of it (based on Result 2 or Result 3) is sufficient. In fact, in the case of streams, there is usually no finite set called population, and the data is generated online.
- A decision can be made about one part of the population independently of other parts.

5 Stratified stream sampling (SSS)

In this section, based on the idea of Result 3, we try to partition the population into windows of integer size and then implement different designs, inside each window. Consider a population with $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)^\top$, and an arbitrary integer number m. Let $k_1 \in U$ be the index such that

$$\sum_{k=1}^{k_1-1} \pi_k < m \text{ and } \sum_{k=1}^{k_1} \pi_k \geq m. \tag{6}$$

Then we split $\pi_{k_1} = \pi_{a_1} + \pi_{b_1}$ such that

$$\sum_{k=1}^{k_1-1} \pi_k + \pi_{a_1} = m.$$

Now, for building the first windows, say w_1, as well as the other windows, we have

$$\boldsymbol{\pi} = (\underbrace{\pi_1, \quad \pi_2, \quad \dots, \quad \pi_{k_1-2}, \quad \pi_{k_1-1}, \quad \pi_{a_1}}_{w_1:\ \text{The first window}} + \pi_{b_1}, \quad \pi_{k_1+1}, \quad \dots, \quad \pi_N)^\top.$$

With the same approaches we build the other windows of size m:

$$\begin{aligned}\boldsymbol{\pi} =& (\underbrace{\pi_1, \pi_2, \dots, \pi_{k_1-2}, \pi_{k_1-1}, \pi_{a_1}}_{w_1,\text{ of size m}} \\ &+ \underbrace{\pi_{b_1}, \pi_{k_1+1}, \dots, \pi_{k_2-1}, \pi_{a_2}}_{w_2,\text{ of size m}} \\ &+ \pi_{b_2}, \dots, \pi_{a_{i-1}} + \underbrace{\pi_{b_{i-1}}, \pi_{k_{i-1}+1}, \dots, \pi_{k_i-1}, \pi_{a_i}}_{w_i,\text{ of size m}\dots} + \pi_{b_i} \dots)^\top.\end{aligned}$$

We call unit $k = k_i$, the cross-border unit of the window w_i. To continue, we first consider the first window. For the first unit inside w_1, its inclusion probability will be updated as (2), and it is possible to update other units inside the first window as (3) for $k = 2, 3, \dots, a_1$ with defining constant c_1 by $\sum_{k=2}^{a_1} \min(c_1\pi_k, 1) = m$. The rest of the population units, stay unchanged as

$$\pi_k^1 = \pi_k; \quad k = b_1, k_1 + 1, k_1 + 2, \dots, N.$$

After deciding on the first unit, to continue, since $\sum_{k=2}^{a_1} \pi_k^1$ is an integer, according to Result 3, we can make a decision on all the units inside the respective window step by step, except maybe for the last one which is a part of a cross-border unit and is not a real unit.

From now on, to simplify notation, π_k^+ and π_k^* denote the final update (which leads to a 0-1 decision) and the initial update (which needs to be imposed on the new window units before entering it) on unit k, respectively. With these notations, after a complete decision about the first window (which will lead to selecting a sample of size m) π_{a_1} will be finally updated to 1 (or 0) with probability π_{a_1} (or $1 - \pi_{a_1}$). Before starting to update the units inside w_2, we first need to make a decision on unit $k = k_1$, and update the other units based on $\pi_{a_1}^+$. For this purpose, we proceed as follows,

- if $\pi^+_{a_1} = 1$, we consider $k = k_1$ as a sample unit ($\pi^*_{k_1} = \pi^+_{k_1} = 1$) and then the extra part in w_2, i.e. π_{b_1}, should be distributed on the other units inside w_2 to initially update them as

$$\pi_k^{*(1)} = \min(c_2^* \pi_k, 1), k = k_1 + 1, \dots, a_2, \text{ where } \sum_{k=k_1+1}^{a_2} \min(c_2^* \pi_k, 1) = m. \tag{7}$$

- if $\pi^+_{a_1} = 0$, we do not decide on the cross-border unit k_1, and the units inside w_2 including b_1 will be updated as

$$\pi_k^{*(0)} = \frac{\pi_k - \pi_k^{*(1)} \pi_{a_1}}{(1 - \pi_{a_1})}, k = b_1, k_1 + 1, \dots, a_2. \tag{8}$$

The Details of the implementation of the method are given in Algorithm 3.

Algorithm 3 SSS with Integer Windows of Size m

```
Initialize $b_0 = 0, k_0 = 1, \pi^*_{a_0} = 0, \pi_0 = 0$,
for $i = 1, 2, \dots$ do,
    Wait until $\sum_{k=b_{i-1}}^{k_i} \pi_k \geq m$, then split $\pi_{k_i} = \pi_{a_i} + \pi_{b_i}$ such that $\sum_{k=b_{i-1}}^{k_i - 1} \pi_k + \pi_{a_i} = m$.
    Calculate $c_i$ such that $\sum_{k=k_{i-1}+1}^{a_i} \min(c_i \pi_k, 1) = m$,
    if $\pi^*_{a_{i-1}} = 1$ then, $\pi^*_{k_{i-1}} = 1$, and
            $$\pi^*_k = \min(c_i \pi_k, 1), k = k_{i-1} + 1, \dots, a_i,$$
        Set $\boldsymbol{\pi}^*_i = (\pi^*_k, k = k_{i-1} + 1, \dots, a_i)$.
    else $\pi^*_{k_{i-1}} = (\pi_{k_{i-1}} - \pi_{a_{i-1}})/(1 - \pi_{a_{i-1}})$, and
            $$\pi^*_k = \frac{\pi_k - \min(c_i \pi_k, 1)\pi_{a_{i-1}}}{(1 - \pi_{a_{i-1}})}, k = k_{i-1} + 1, \dots, a_i,$$
        Set $\boldsymbol{\pi}^*_i = (\pi^*_k, k = k_{i-1}, \dots, a_i)$.
    end if
    Implement OSOD or any other fixed size design on $\boldsymbol{\pi}^*_i$.
end for
```

Result 4 shows that SSS respects the first-order inclusion probability and fixed sample size.

Result 4 In SSS, for $k = 1, 2, \dots$ and $i = 1, 2, \dots$,

(i) $E(\pi^*_k) = \pi_k$,

(ii) $0 \leq \pi_k^{*(0)}, \pi_k^{*(1)} \leq 1$,

(iii) $\sum_{k=b_i}^{a_{i+1}} \pi_k^{*(0)} = m$, and $\sum_{k=b_i}^{a_{i+1}} \pi_k^{*(1)} = m - 1$.

Since the size of the windows is an integer, we can implement any arbitrary design and sample size inside each stratum. Different designs and sample sizes inside different strata make the design completely flexible. Based on the next result, we can calculate all the second-order inclusion probabilities which makes SSS a flexible stream sampling design, capable to estimate the accuracy of the estimations.

Result 5 With m as the size of the windows, considering p_i as the sampling design implemented in i^{th} window, and defining

$$f(k, d_1, c, d_2) = \pi_{d_1|k} \min(c\pi_{d_2}, 1) + (1 - \pi_{d_1|k}) \frac{\pi_{d_2} - \min(c\pi_{d_2}, 1)\pi_{d_1}}{1 - \pi_{d_1}},$$

and

$$\pi_{d_1 d_2}^{p_i(a_i)} = \pi_{a_i} \pi_{d_1 d_2}^{p_i(a_i \in S)} + (1 - \pi_{a_i}) \pi_{d_1 d_2}^{p_i(a_i \notin S)}$$

where $\pi_{d_1 d_2}^{p_i}$, $\pi_{d_1 d_2}^{p_i(a_i \in S)}$ and $\pi_{d_1 d_2}^{p_i(a_i \notin S)}$ are the second-order inclusion probability of units d_1 and d_2 with implementing design p_i in window i, when they are updated given $\pi_{a_i}^{+} = 1$ similar to (7) and they are updated given $\pi_{a_i}^{+} = 0$ according to (8) respectively, we have

(i) if k and ℓ are two non-cross-border units belonging to the same window, say i, then

$$\pi_{k\ell} = \begin{cases} \pi_{k\ell}^{p_i}, & i = 1 \\ \pi_{k\ell}^{p_i(a_i)}, & i > 1. \end{cases}$$

(ii) if k and ℓ are two non-cross-border units belonging to distinct windows i and j, respectively, where $i < j$ then,

$$\pi_{k\ell} = \pi_k f(k, a_{j-1}, c_j, \ell),$$

where $\pi_{a_{j-1}|k}$ can be calculated based on a recursive relation as

$$\pi_{a_{j'}|k} = \begin{cases} f(k, a_{j'-1}, c_{j'}, a_{j'}), & i < j' < j \\ \dfrac{\pi_{ka_i}^{p_i(a_{i-1})}}{\pi_k} & j' = i. \end{cases}$$

(iii) if $k = k_i$ and ℓ is a non-cross-border belonging to window j, where $i < j$ then

$$\pi_{k\ell} = \pi_k f(k, a_{j-1}, c_j, \ell),$$

where

$$\pi_{a_{j'}|k} = \begin{cases} f(k, a_{j'-1}, c_{j'}, a_{j'}), & i+1 < j' < j \\ \pi_{a_i|k} \min(c_{i+1}\pi_{a_{i+1}}, 1) + (1-\pi_{a_i|k}) \frac{\pi^{p_i(a_i \notin s)}_{b_i a_{i+1}}}{\pi_{b_i}/(1-\pi_{a_i})}, & j' = i+1, \\ \frac{\pi_{a_i}}{\pi_k} & j' = i. \end{cases}$$

(iv) if $\ell = k_j$ and k is a non-cross-border unit belonging to window i, where $i < j$ then

$$\pi_{k\ell} = \pi_k \left[\pi_{a_j|k} + (1-\pi_{a_j|k}) \frac{\pi_{b_j}}{1-\pi_{a_j}} \right],$$

where

$$\pi_{a_{j'}|k} = \begin{cases} f(k, a_{j'-1}, c_{j'}, a_{j'}), & i < j' < j \\ \frac{\pi^{p_i(a_{i-1})}_{ka_i}}{\pi_k}, & j' = i > 1 \\ \frac{\pi^{p_i}_{ka_i}}{\pi_k} & j' = i = 1. \end{cases}$$

(v) if $k = k_i$ and $\ell = k_j$, where $i < j$, we have

$$\pi_{k\ell} = \pi_k \pi_{\ell|k},$$

then

$$\pi_{k|\ell} = \left[\pi_{a_j|k} + (1-\pi_{a_j|k}) \frac{\pi_{b_j}}{1-\pi_{a_j}} \right],$$

$$\pi_{a_{j'}|k} = \begin{cases} f(k, a_{j'-1}, c_{j'}, a_{j'}), & i+1 < j' \leq j \\ \frac{\pi_{a_i}}{\pi_k} \min(c_j \pi_{a_j}, 1) + (1 - \frac{\pi_{a_i}}{\pi_k}) \frac{\pi^{p_i(a_i \notin s)}_{b_i a_j}}{\pi_{b_i}/(1-\pi_{a_i})}, & j' = i+1. \end{cases}$$

In addition to proving Result 5 in Appendix, we will run some simulations to evaluate calculation of the second-order inclusion probabilities in Sect. 8.

6 Immediate decision sampling (IDS)

Here we show that the method of windows of size one leads to immediate decision and is equivalent to Chromy sequential method, Deville's systematic sampling, and the order pivotal method. Also, we will present a very simple algorithm for the method, with only one condition.

If we set $m = 1$ in (6), then we will have windows of size one. Since the sum of inclusion probabilities within the window is one, if a unit is selected, all the other inclusion probabilities will be updated to zero to compensate for the required inclusion probabilities, i.e. $\pi_k^{1(1)} = 0$ in (3) while $\pi_k^{1(0)} = \pi_k/(1-\pi_k)$. Result 6 shows that setting $m = 1$, will lead to an immediate decision sampling on the population units.

Result 6 With $F_0 = 0$ and $F_\ell = \sum_{k=1}^{\ell} \pi_k$, windows of size one is equivalent to the following process:

After observing unit ℓ in windows w_i, if

(I) ℓ is not a cross-border unit,

(1) if $n_\ell < i$,

$$\pi_\ell^* = \frac{\pi_\ell}{1-(F_{\ell-1}-\lfloor F_{\ell-1}\rfloor)},$$

(2) if $n_\ell = i$,

$$\pi_\ell^* = 0.$$

(II) ℓ is a cross-border unit,

(1) if $n_\ell < i$,

$$\pi_\ell^* = 1,$$

(2) if $n_\ell = i$,

$$\pi_\ell^* = \frac{\pi_\ell - \left\{1-(F_{\ell-1}-\lfloor F_{\ell-1}\rfloor)\right\}}{F_{\ell-1}-\lfloor F_{\ell-1}\rfloor}.$$

Then, based on IDS, there is no need to know about unobserved units coming in the future to make a decision on the observed unit. The four IDS scenarios in Result 6, can be summarized in one condition as presented in Algorithm 4.

Algorithm 4 Immediate Decision Sampling

```
Initialize with $F = 0$, $n = 0$ and $s = \{\}$,
After receiving unit $\ell$ in a stream, follow the steps below,
 (1) Generate $u$, a realization of a uniform random variable in $[0, 1]$,
 (2) Set $F_1 = F$, $F = F + \pi_\ell$, $\alpha = \lceil F \rceil - \lfloor F_1 \rfloor - 1$, $\beta = \lceil F \rceil - n$ and $m = F_1 - \lfloor F_1 \rfloor$
 (3) If
         $$u \leq \min(\beta, 1) \frac{\pi_\ell - \alpha(2-\beta)(1-m)}{(1-\alpha)(1-m) + \alpha\{(2-\beta)m + \pi_\ell(\beta - 1)\}}$$
     then
         $$s = s \cup \ell, n = n + 1.$$
```

Result 7 IDS, Chromy sequential method, Deville's systematic sampling, and the order pivotal method are four implementations of the same design.

It is worth noting that although IDS and Chromy sequential method are the same method, but in Algorithm 4, we have provided a very easy algorithm to code the design with only one condition. Then, with a fixed order of population units, IDS can be considered as a simpler version of Deville's method. In IDS, according to Algorithm 4, there is no need to check whether a unit is a cross-border unit or not, and there is no need to generate random variables from different distributions for each window.

7 Size of the windows, spreading the sample, and the designs inside

There are a few points about window size. Regarding the optimal window size, in the case of streaming data, it is preferable to decide on the current units as soon as possible. In Sect. 6, we showed that if we set the window sizes to one, the method is equivalent to Chromy (1979), which is an immediate decision sampling for each observed unit. As a drawback, windows of size one lead to many zero second-order inclusion probabilities. In fact, the second-order inclusion probabilities are zero among all the units completely inside each window. A solution investigated by Chauvet (2021), is to change the starting unit randomly. This approach contrasts with the data streaming aspect. Then the balance between having (almost) all the second-order inclusion probabilities positive, respecting the streaming aspect and immediate decision leads to set $m = 2$. For this purpose, after receiving enough data to make a window of size $m = 2$, we can completely decide on the data inside the window.

Regarding the feasibility of implementing the method, if we are dealing with a streaming population, we will always have enough data to make the current window of size m. To implement the method inside a finite population, one can decide on m according to the population size and inclusion probabilities. For the last window, we may have to resize the window and if the sum of the inclusion probabilities of the last window is not an integer, we can solve the problem by using a phantom unit (Grafström et al. 2012).

Moreover, if the population has spatial coordinates, we can construct windows based on the coordinates. Then, the smaller the windows size, the spreader the sample is in the population coordinates, which can lead to more efficient samples (see Grafström and Lundström 2013).

8 Examples and simulations

In this section, the validity of calculating second-order inclusion probabilities and efficiency of the proposed design will be evaluated using several examples and simulations.

8.1 Evaluating result 5

To have a general case, consider a population of size $N = 15$ with $n = 7$ and the strata size $H = 3$ with

$$\boldsymbol{\pi} = (.34, .82, .47, .66, .47, .36, .40, .53, .20, .22, .81, .67, .53, .11, .41). \tag{9}$$

To have strata of size $H = 3$, since the total size is $n = 7$, we will have an unbalanced stratified population. The population is partitioned in $L = 3$ strata, all size $H_1 = H_2 = 3$ but the last one of size $H_3 = 1$ (see Table 1).

Let matrix $\mathbf{\Pi}$ be the second-order inclusion probabilities calculated based on Result 5,

Table 1 Stratification of population (9) with $N = 15$, $n = 7$, stratified in $L = 3$ windows of sizes $H_1 = 3$, $H_2 = 3$ and $H_3 = 1$ respectively

$L = 3$	$i = 1$	$i = 2$	$i = 3$
k_i	6	13	15
H_i	3	3	1
π_{k_i}	0.36	0.53	0.41
π_{a_i}	0.24	0.05	0.41
π_{b_i}	0.12	0.48	0.00

$$\mathbf{\Pi}=\begin{bmatrix}
.34 & .26 & .11 & .18 & .11 & .08 & .14 & .18 & .07 & .07 & .28 & .23 & .18 & .04 & .14\\
 & .82 & .35 & .49 & .35 & .28 & .33 & .43 & .17 & .18 & .66 & .55 & .43 & .09 & .33\\
 & & .47 & .25 & .15 & .13 & .19 & .24 & .09 & .10 & .38 & .31 & .25 & .05 & .19\\
 & & & .66 & .25 & .21 & .26 & .34 & .13 & .14 & .53 & .44 & .35 & .07 & .27\\
 & & & & .47 & .14 & .19 & .25 & .10 & .10 & .38 & .32 & .25 & .05 & .19\\
 & & & & & .36 & .13 & .18 & .06 & .06 & .29 & .24 & .19 & .04 & .15\\
 & & & & & & .40 & .15 & .05 & .05 & .30 & .22 & .21 & .04 & .16\\
 & & & & & & & .53 & .08 & .08 & .39 & .29 & .28 & .06 & .21\\
 & & & & & & & & .20 & .01 & .15 & .11 & .10 & .02 & .08\\
 & & & & & & & & & .22 & .16 & .12 & .11 & .02 & .09\\
 & & & & & & & & & & .81 & .50 & .43 & .09 & .33\\
 & & & & & & & & & & & .67 & .35 & .07 & .27\\
 & & & & & & & & & & & & .53 & .01 & .04\\
 & & & & & & & & & & & & & .11 & .00\\
 & & & & & & & & & & & & & & .41
\end{bmatrix}$$

and matrix $\widehat{\mathbf{\Pi}}$ be the second-order inclusion probabilities based on a Monte Carlo simulation with $M = 10,000$ iterations. In each iteration, we implemented SSS in Sect. 5 using `R` where within the stratum elimination procedure (Tillé 1996) by function `UPtille` under library `sampling` had been used, then 10, 000 samples have been obtained and the element (k,ℓ) in the matrix $\widehat{\mathbf{\Pi}}$ indicates number of times that the units k and ℓ have been selected together in the same sample, resulted in:

$$\hat{\mathbf{\Pi}}=\begin{bmatrix}
.35 & .26 & .11 & .19 & .11 & .08 & .14 & .18 & .07 & .08 & .28 & .23 & .18 & .04 & .15\\
 & .81 & .34 & .49 & .35 & .28 & .32 & .42 & .17 & .18 & .67 & .55 & .43 & .09 & .33\\
 & & .46 & .25 & .15 & .13 & .18 & .24 & .10 & .09 & .38 & .31 & .24 & .05 & .19\\
 & & & .66 & .25 & .21 & .26 & .35 & .14 & .14 & .54 & .45 & .35 & .07 & .27\\
 & & & & .47 & .14 & .19 & .24 & .09 & .10 & .38 & .31 & .25 & .05 & .19\\
 & & & & & .36 & .13 & .17 & .06 & .06 & .29 & .23 & .19 & .04 & .14\\
 & & & & & & .40 & .14 & .04 & .05 & .30 & .22 & .21 & .04 & .16\\
 & & & & & & & .52 & .07 & .08 & .39 & .30 & .27 & .06 & .21\\
 & & & & & & & & .21 & .01 & .16 & .11 & .11 & .02 & .08\\
 & & & & & & & & & .21 & .16 & .12 & .11 & .02 & .08\\
 & & & & & & & & & & .82 & .51 & .43 & .09 & .34\\
 & & & & & & & & & & & .67 & .35 & .07 & .27\\
 & & & & & & & & & & & & .53 & .01 & .04\\
 & & & & & & & & & & & & & .11 & .00\\
 & & & & & & & & & & & & & & .41
\end{bmatrix}.$$

The comparison of $\mathbf{\Pi}$ and $\widehat{\mathbf{\Pi}}$ confirms the validity of Result 5. As we can see in $\mathbf{\Pi}$, all the second-order inclusion probabilities are positive, except for $\pi_{k\ell}$ with $k = 14, \ell = 15$ which was obvious due to the sample size $H_3 = 1$ in the third stratum.

Also, for larger population sizes, $N = 100, 200$, with different sample sizes and different strata sizes, we run Monte Carlo simulations with $M = 10,000$ iterations. The first-order inclusion probabilities were set putting random uniform distribution into the `inclusionprobabilities` function under the `sampling` library. Again, within each stratum, elimination procedure was used to select a sample. To compare the calculated ($\mathbf{\Pi}$, based on Result 5) and the estimated ($\widehat{\mathbf{\Pi}}$, based on Monte Carlo) matrices of the second-order inclusion probabilities, we defined the following criterion

$$\mathcal{Z}_{\pi}=\frac{\sum_{k=1}^{N}\mathcal{I}\left(\left|\frac{\hat{\pi}_{kk}-\pi_{k}}{\sqrt{\pi_k(1-\pi_k)/M}}\right|>1.96\right)}{N},\qquad \mathcal{Z}_{\diamond}=\frac{\sum\sum_{k\le\ell}^{N}\mathcal{I}\left(\left|\frac{\hat{\pi}_{k\ell}-\pi_{k\ell}}{\sqrt{\pi_{k\ell}(1-\pi_{k\ell})/M}}\right|>1.96\right)}{\frac{N(N-1)}{2}},$$

where $\pi_{k\ell}$ and $\hat{\pi}_{k\ell}$ are the (k, l) elements of $\mathbf{\Pi}$ and $\widehat{\mathbf{\Pi}}$ respectively. Also, $\mathcal{I}(.)$ is an indicator function that takes 1 if "." is satisfied. Based on Central Limit Theorem, we accept the validity of the results with a confidence level of 0.95 if $\mathcal{Z} \le 0.05$.

Table 2 Results for 10,000 Monte Carlo iterations of implementing the SSS design in random populations of sizes $N = 200, 100$

N	n	H	L	$\mathcal{Z}_\pi$	$\mathcal{Z}_\diamond$
200	20	5	4	0.05	0.05
	30		6	0.05	0.05
	50		10	0.06	0.05
100	15	3	5	0.05	0.05
	30		7	0.05	0.04
	50		10	0.04	0.05

Results are shown in Table 2 which once again confirms the validity of Result 5 independent of N, n, H, and L.

8.2 Efficiency of SSS

The R package library `datasets` (R Core Team 2022) contains the `swiss` database. Switzerland, in 1888, was entering a period known as the demographic transition; i.e., its fertility was beginning to fall from the high level typical of underdeveloped countries. The data concerns $N = 47$ French-speaking districts at about this period of the demographic transition.

We are convinced that SSS is particularly efficient in the research field of stream sampling. To evaluate the accuracy of the SSS estimator relative to other sampling design like elimination and max entropy, we used `swiss` data, including 3 variables (see Fig. 1):

- Fertility: common standardized fertility measure,
- Agriculture: % of males involved in agriculture as occupation,
- Infant Mortality: live births who live less than 1 year.

"Fertility" variable was used for the main variable y and "Agriculture" and "Infant Mortality" variables were considered for auxiliary variables to create first-order inclusion probabilities. Also, we defined the efficiency of SSS as

$$\text{EF}(.) = \frac{Var_M(\hat{Y}_.)}{Var_M(\hat{Y}_{\text{SSS}})},$$

where "." indicates "elimination" or "max entropy" design, and "Var_M" indicates the Monte Carlo variance of the respective estimators based on 10,000 iterations. Results are shown in Table 3 which shows that SSS is quite comparable with other commonly used sampling designs in terms of accuracy. The smaller the size of the strata and the larger the sample size, the more efficient the SSS. As an interesting point, for all cases with $m = 1$, SSS is more efficient than any other methods.

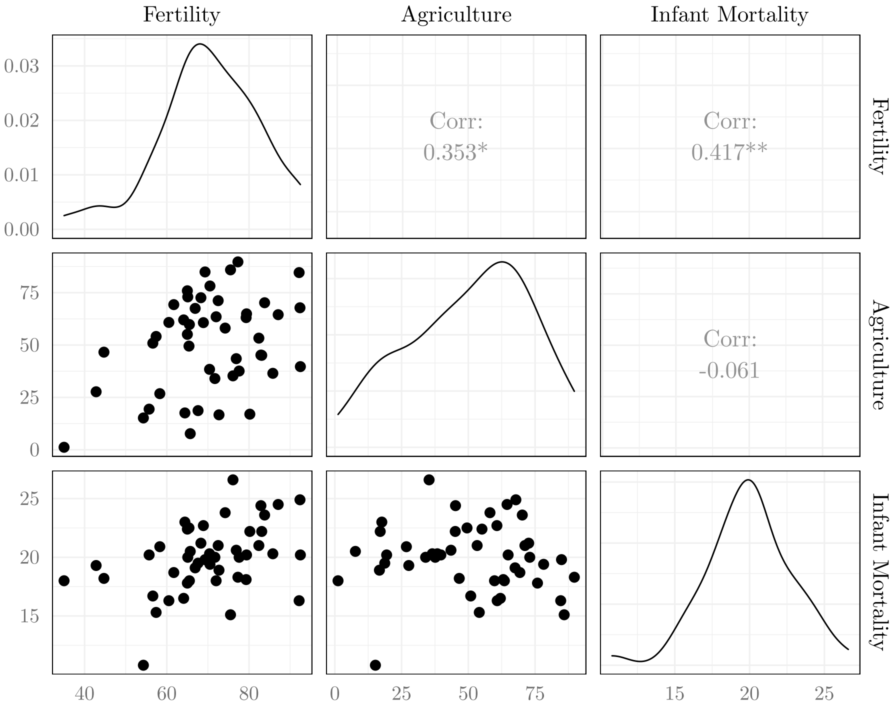

Fig. 1 The distribution and relationship among three variables of interest, Fertility, Agriculture and Infant Mortality, for the `swiss` data. In this 3 × 3 matrix of plots, the lower off-diagonal draws scatter plots, the diagonal represents density plots and the upper off-diagonal reports the Pearson correlations

Table 3 Efficiency of SSS relative to elimination and maximum entropy designs based on Monte Carlo variance of the estimators

n	*H*	Fertility—agriculture		Fertility—infant.mortality	
		EF(elimination)	EF(max entropy)	EF(elimination)	EF(max entropy)
6	1	1.07	1.11	1.20	1.21
	2	0.98	1.02	0.98	0.99
	3	0.99	1.02	0.98	0.98
9	1	1.14	1.21	1.16	1.17
	2	1.11	1.18	1.18	1.19
	3	0.98	1.04	0.98	0.99
12	1	1.15	1.24	1.26	1.27
	2	1.05	1.13	1.20	1.21
	3	1.00	1.09	0.98	0.99

9 Conclusion

SSS is a general stream sampling method that allows decisions on units based on their order. The order can be arbitrary, but if the data are the result of a stream, this method can be very useful for deciding which units should be stored as samples over time. Then, after deciding on each unit to update the inclusion probabilities, it is not necessary to use all the population, and usually only a small window of the population can be sufficient. SSS is flexible for stratifying populations and allocating samples into the strata, leading to almost immediate decision on units. In spatial data it is possible to spread the sample over the population coordinates, while the level of spreading can be controlled using window sizes. Window size is a leverage that can be used to make a trade-off between design entropy and sample dispersion over the population indices. The smaller the window size, the greater the sample dispersion the lower the entropy of the design, and vice versa. Size one and two windows are special cases with interesting properties. The former leads to an immediate decision where after observing each unit we can make an immediate decision about its selection without depending on unobserved units. On the other hand, the latter is one of the optimal options, which spreads the sample over the population indices with a reasonable entropy that gives positive chance to all second-order inclusion probabilities.

For future research, it will be interesting to investigate the problem of spreading the sample over the population coordinates using SSS and compare with other successful methods in spatial spreading of sample units, such as local pivotal (Grafström et al. 2012) and weakly associated vectors (Jauslin and Tillé 2020) methods. For this purpose, in two-dimensional coordinates, one unit can be considered as the center, and then using the units close to build the strata. Choosing central units is one of the challenges of this method. Such units can be selected randomly or non-randomly. Since the windows of size one, splits one of the inclusion probabilities to make the smallest possible strata, the level of spreading would probably be good and worth studying.

Appendix

Proof of Result 1 Without loss of generality and for ease of notation we only do the proof for $t = 1$. For $k = 1$ it is obvious that $\mathrm{E}(\pi_1^1) = \pi_1$, and for $k = 2, 3, \dots, N$,

$$\mathrm{E}(\pi_k^1) = \pi_k^{1(0)}(1 - \pi_1) + \pi_k^{1(1)}\pi_1 = \pi_k^{1(0)}(1 - \pi_1) + \frac{\pi_k - \pi_k^{1(0)}(1 - \pi_1)}{\pi_1}\pi_1 = \pi_k.$$

Also for the sum of inclusion probabilities, we have

$$\sum_{k\in U} \pi_k^{1(0)} = 0 + \sum_{k=2}^{N} \pi_k^{1(0)} = \sum_{k=2}^{N} \min(c_1\pi_k, 1) = n = \sum_{k=1}^{N} \pi_k,$$

and

$$\begin{aligned}\sum_{k\in U}\pi_k^{1(1)} &= 1+\sum_{k=2}^{N}\pi_k^{1(1)}\\ &= 1+\frac{\sum_{k=2}^{N}\pi_k-(1-\pi_1)\sum_{k=2}^{N}\pi_k^{1(0)}}{\pi_1}\\ &= \frac{\pi_1+\{(n-\pi_1)-n+n\pi_1\}}{\pi_1}=n=\sum_{k=1}^{N}\pi_k.\end{aligned}$$

□

Proof of Result 2 Without loss of generality and for ease of notation we only do the proof for $t=1$. Since

$$\pi_k^{1(1)}=\begin{cases}\dfrac{\pi_k-\min(c_1\pi_k,1)(1-\pi_1)}{\pi_1}\leq\dfrac{\pi_k-c_1\pi_k(1-\pi_1)}{\pi_1} & \text{if } \pi_k^{1(1)}=1\\ \dfrac{\pi_k-c_1\pi_k(1-\pi_1)}{\pi_1} & \text{if } \pi_k^{1(1)}<1,\end{cases}$$

for $k=2,3,\ldots,N$, a necessary and sufficient condition is that $\pi_1\geq 1-1/c_1$, which gives the result. □

Proof of Result 3 Let define

$$\begin{aligned}U_A=&\{k\in U|0<\pi_k^{1(0)}<1\},\ U_B=\{k\in U|\pi_k^{1(0)}=1\},\\ A=&\sum_{k\in U_A}\pi_k \text{ and } B=\sum_{k\in U_B}\pi_k.\end{aligned}$$

Then it is possible to decompose n as

$$\pi_1+A+B=n \tag{10}$$

and

$$c_1A+\#U_B=n,$$

where “#” indicates cardinality. Now, if n is an integer, c_1A is an integer denoted by d, thus $A=d/c_1$. In this case, we have $\#U_B=n-d$. Furthermore, it is easy to see that $B\leq\#U_B$. Now, from Eq. (10), we have

$$\pi_1=n-A-B=n-\frac{d}{c_1}-B\geq n-\frac{d}{c_1}-\#U_B=n-\frac{d}{c_1}-(n-d)=d\left(1-\frac{1}{c_1}\right).$$

Now if $d=1,2,\ldots$ then $\pi_1\geq(1-1/c_1)$ and if $d=0$ then $\#U_B=n$ or in other words, $\pi_k^{1(0)}=1$ for all $k=2,3,\ldots,N$. Therefore, to have all $\pi_k^{1(1)}\geq 0$, it is necessary to have

$$\pi_k^{1(1)}=\frac{\pi_k-(1-\pi_1)}{\pi_1}\geq 0,\Rightarrow \pi_k+\pi_1-1\geq 0. \tag{11}$$

But as in such cases, $\pi_k^{1(0)} = \pi_k + \alpha_k \pi_1 = 1$, for some $0 < \alpha_k \leq 1$, then $\pi_k + \pi_1 \geq 1$ and therefore Condition (11) is satisfied.

Then, if n is an integer, the condition of Result 2 is always fulfilled. □

Proof of Result 4 Proof for w_1 is obvious based on Result 1. We prove the result for w_2:

(i) For $k = k_1 + 1, \dots, a_2$,

$$E(\pi_k^*) = \pi_{a_1}\pi_k^{*(1)} + (1 - \pi_{a_1})\pi_k^{*(0)} = \pi_{a_1}\pi_k^{*(1)} + \pi_k - \pi_{a_1}\pi_k^{*(1)} = \pi_k,$$

and for $k = k_1$,

$$E(\pi_{k_1}^*) = \pi_{a_1} \times 1 + (1 - \pi_{a_1})\frac{\pi_{k_1} - \pi_{a_1}}{(1 - \pi_{a_1})} = \pi_{k_1}.$$

(ii) For $\pi_{a_1}^+ = 1$, in (7), actually π_{b_1} will be distributed on $\pi_{k_1+1}, \dots, \pi_{a_2}$. To show that the inclusion probabilities in (8) are non-negative, we have

$$\frac{\pi_k - \min(c_2^*\pi_k, 1)\pi_{a_1}}{(1 - \pi_{a_1})} \geq \frac{\pi_k - c_2^*\pi_k\pi_{a_1}}{(1 - \pi_{a_1})} \geq 0$$

which leads to

$$(1 - \pi_{a_1}) \geq (1 - \frac{1}{c_2^*}).$$

But the size of w_2 is an integer, we know that

$$\frac{\pi_k - \min(c_2^*\pi_k, 1)(1 - \pi_{b_1})}{\pi_{b_1}} \geq \frac{\pi_k - c_2^*\pi_k(1 - \pi_{b_1})}{\pi_{b_1}} \geq 0$$

and then

$$\pi_{b_1} \geq \left(1 - \frac{1}{c_2^*}\right).$$

Therefore, as $\pi_{a_1} + \pi_{b_1} = \pi_{k_1} \leq 1$ we have

$$(1 - \pi_{a_1}) \geq \pi_{b_1} \geq \left(1 - \frac{1}{c_2^*}\right).$$

(iii) Proof for respecting sum of the inclusion probabilities are straightforward by calculating summation of $\pi_k^{*(0)}$ and $\pi_k^{*(1)}$ inside w_2.

For the other windows, proof is the same. □

Proof of Result 5 For calculating the second-order inclusion probability $\pi_{k\ell}$ where $k < \ell$, $k \in w_i$ and $\ell \in w_j$, we have

$$\Pr(k \in S, \ell \in S) = \Pr(k \in S)\Pr(\ell \in S \mid k \in S) = \pi_k \pi_{\ell|k}.$$

In $\pi_{\ell|k}$, given k is selected affect on selecting π_ℓ by changing $\pi_{a_{j-1}}$. Then based on a recursive relation, step by step we can calculate $\pi_{a_{j-1}|k}$ using $\pi_{a_{j-2}|k}$ and so on. Then we need to consider the cases in Result 5, as

(i) in this case, the second inclusion probabilities can be calculated based on the design, p_i, implemented inside the respective window,

(ii) here, after following recursive calculation for calculating $\pi_{a_i|k}$, as a_i and k are in the same window, we have

$$\pi_{a_i|k} = \frac{\pi_{ka_i}}{\pi_k} = \frac{\pi_{ka_i}^{p_i}}{\pi_k},$$

(iii) here, since unit k is a cross-border unit, if a_i is selected, $\pi_{a_{i+1}}$ will be updated as $min(c_{i+1}\pi_{a_{i+1}}, 1)$, and if a_i is not selected, then $\pi_{a_{i+1}}$ will be updated as

$$\frac{\pi_{b_i a_{i+1}}^{p_i(a_i \ni S)}}{\pi_{b_i}/(1-\pi_{a_i})}.$$

For conditional probability of a_i itself, as a_i is a part of $\pi_k = a_i + b_i$, and then

$$\{a_i \in S\} \subset \{k \in S\},$$

therefore we have

$$\pi_{a_i|k} = \frac{\Pr(k \in S, a_i \in S)}{\pi_k} = \frac{\Pr(a_i \in S)}{\pi_k} = \frac{\pi_{a_i}}{\pi_k},$$

(iv) when $\pi_\ell = \pi_{a_j} + \pi_{b_j}$, then

$$\pi_{\ell|k} = \Pr(a_j \in S) + \Pr(a_j \notin S)\Pr(b_j \in S \mid a_j \notin S) = \pi_{a_j|k} + (1 - \pi_{a_j|k})\frac{\pi_{b_j}}{1-\pi_{a_j}},$$

and the rest of the proof is the same as case *ii*),

(v) the first part of the proof of this case is the same as case *iv*), and after recursive calculations, the last part is the same as the last part of case *iii*).

□

Proof of Result 6 After deciding on the first window, as a_1 is not a real unit, depending on the decision for this unit, the units inside w_2 will be initially updated as

$$\pi^*_{b_1} = \begin{cases} \pi^{*(1)}_{b_1} = 0 & \text{if } \pi^+_{a_1} = 1 \\ [2mm]\pi^{*(0)}_{b_1} = \dfrac{\pi_{k_1} - \pi_{a_1}}{1 - \pi_{a_1}} & \text{if } \pi^+_{a_1} = 0, \end{cases} \tag{12}$$

and

$$\pi^*_k = \begin{cases} \pi^{*(1)}_k = \frac{\pi_k}{1-\pi_{b_1}} & \text{if } \pi^+_{a_1} = 1 \\ [2mm]\pi^{*(0)}_k = \dfrac{\pi_k - \pi^{*(1)}_k \pi_{a_1}}{1 - \pi_{a_1}} & \text{if } \pi^+_{a_1} = 0, \end{cases} \quad \text{for } k = k_1 + 1, \ldots, a_2. \tag{13}$$

Consider unit ℓ, inside w_2

(I) j is not a cross-border unit,

(1) if $n_\ell < i$ and $\pi^+_{a_1} = 1$, then according to (13) we have

$$\pi^*_\ell = \frac{\frac{\pi_\ell}{1-\pi_{b_1}}}{1 - \sum_{i=k_1+1}^{\ell-1} \frac{\pi_\ell}{1-\pi_{b_1}}} = \frac{\pi_\ell}{1 - (F_{\ell-1} - \lfloor F_{\ell-1} \rfloor)},$$

and if $\pi^+_{a_1} = 0$,

$$\pi^*_\ell = \frac{\frac{\pi_\ell - \pi^{*(1)}_\ell \pi_{a_1}}{1-\pi_{a_1}}}{1 - \frac{\pi_{k_1} - \pi_{a_1}}{1-\pi_{a_1}} - \sum_{i=k_1+1}^{\ell-1} \frac{\pi_\ell - \pi^{*(1)}_\ell \pi_{a_1}}{1-\pi_{a_1}}},$$

which with replacing $\pi^{*(1)}_\ell$ by $\pi_\ell/(1 - \pi_{b_1})$ we have

$$\pi^*_\ell = \frac{\pi_\ell}{1 - (F_{\ell-1} - \lfloor F_{\ell-1} \rfloor)}.$$

(2) if $n_\ell = i$, according the window size, this unit will not be selected.

(II) ℓ is a cross-border unit,

(1) if $n_\ell < i$, according to the window size, this unit will be selected with probability one.
(2) if $n_\ell = i$, we can calculate π^*_ℓ directly using (12).

For the other windows, the proof is the same.

□

Proof of Result 7 The structure of the population and cross-units inside both methods are the same, and the updating principle of Deville's method is the same as Eqs. (12) and (13). In Deville's method, consider the two first windows,

(I) ℓ is not a cross-border unit,

(1) If the cross-border unit is selected inside the previous window, then

$$\begin{aligned}\Pr(\ell \in S) &= \int_{F_{\ell-1}}^{F_\ell} f(x)dx = \int_{F_{\ell-1}}^{F_\ell} \frac{1}{\lceil F_{k_1} \rceil - F_{k_1}} dx = \frac{1}{\lceil F_{k_1} \rceil - F_{k_1}} \pi_\ell \\ &= \frac{1}{1-\pi_{b_1}} \pi_\ell\end{aligned}$$

which is equivalent to the first term of (13),

(2) If the cross-border unit is not selected inside the previous window, then

$$\begin{aligned}\Pr(\ell \in S) &= \int_{F_{\ell-1}}^{F_\ell} 1 - \frac{(\lceil F_{k_1-1} \rceil - F_{k_1-1})(F_{k_1} - \lfloor F_{k_1} \rfloor)}{\left\{1 - (\lceil F_{k_1-1} \rceil - F_{k_1-1})\right\}\left\{1 - (F_{k_1} - \lfloor F_{k_1} \rfloor)\right\}} dx \\ &= \frac{1-\pi_{k_1}}{(1-\pi_{a_1})(1-\pi_{b_1})} \pi_\ell\end{aligned}$$

which is equivalent to the second term of (13),

(II) ℓ is a cross-border unit ($\ell = k_1$),

(1) If the cross-unit is selected inside the previous window, then the method ignores the second part of k_1, i.e. ($\pi^*_{a_1}$=0), which is equivalent to the first term of (12),

(2) If the cross-unit is not selected inside the previous window, then

$$\begin{aligned}\Pr(\ell \in S) &= \int_{\lfloor F_\ell \rfloor}^{F_\ell} \frac{1}{1 - (\lceil F_{k_1-1} \rceil - F_{k_1-1})} dx \\ &= \frac{1}{1 - (\lceil F_{k_1-1} \rceil - F_{k_1-1})} (F_\ell - \lfloor F_\ell \rfloor) \\ &= \frac{1}{1-\pi_{a_1}} \pi_{b_1} = \frac{\pi_{k_1} - \pi_{a_1}}{1-\pi_{a_1}}\end{aligned}$$

which is equivalent to the second term of (12),

For the other windows, the proof is the same.

Now if s is a fixed sample and $p_I(.)$ and $p_D(.)$ are the designs of IDS and Deville's method respectively, as all the units inside s have to be selected under the same principle in both method, then $p_I(s) = p_D(s)$. Furthermore, it is proved in Chauvet (2012) and Chauvet (2021) that the Deville's, Chromy sequential and order pivotal methods lead to the same design, and then the proof is complete. □

References

Aubry P (2023) On the correct implementation of the hanurav-vijayan selection procedure for unequal probability sampling without replacement. Commun Stat-Simul Comput 52(5):1849–1877

Boley M, Lucchese C, Paurat D, Gartner T (2011) Direct local pattern sampling by efficient two-step random procedures. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD'11, San Diego, USA, 21–24 August 2011. ACM Press, New York, USA, pp 582–590

Busnel Y, Tillé Y (2020) Attack-tolerant unequal probability sampling methods over sliding window for distributed streams. In: 4th international conference on compute and data analysis (ICCDA 2020), Mar 2020, San Jose, United States, pp 72–78

Chao M-T (1982) A general purpose unequal probability sampling plan. Biometrika 69:653–656

Chaudhuri A, Pal S (2022) Sampling with Varying Probabilities. Springer Nature Singapore, Singapore, pp 43–109

Chauvet G (2012) On a characterization of ordered pivotal sampling. Bernoulli 18(4):1320–1340

Chauvet G (2021) A note on chromy's sampling procedure. J Surv Stat Methodol 9(5):1050–1061

Chauvet G (2022) A Cautionary Note on the Hanurav-Vijayan Sampling Algorithm. J Surv Stat Methodol 10(5):1276–1291

Chromy JR (1979) Sequential sample selection methods. In: Proceedings of the American statistical association, survey research methods section, pp 401–406

Cohen E, Duffield N, Kaplan H, Lund C, Thorup M (2009) Stream sampling for variance-optimal estimation of subset sums. In: Proceedings of the twentieth annual ACM-SIAM symposium on discrete algorithms. society for industrial and applied mathematics, pp 1255–1264

Deville J-C, Tillé Y (1998) Unequal probability sampling without replacement through a splitting method. Biometrika 85:89–101

Diop L, Diop CT, Giacometti A, Li D, Soulet A (2018) Sequential pattern sampling with norm constraints. In: t2018 IEEE international conference on data mining (ICDM), pp 89–98

Gabler S (1990) Minimax Solutions in Sampling from Finite Populations. Springer, New York

Giacometti A, Soulet A (2021) Reservoir pattern sampling in data streams. In: Oliver N, Pérez-Cruz F, Kramer S, Read J, Lozano JA (eds) Machine learning and knowledge discovery in databases. Research track. Springer International Publishing, Cham, pp 337–352

Grafström A, Lundström NLP (2013) Why well spread probability samples are balanced? Open J Stat 3(1):36–41

Grafström A, Lundström NLP, Schelin L (2012) Spatially balanced sampling through the pivotal method. Biometrics 68(2):514–520

Grafström A, Matei A, Qualité L, Tillé Y (2012) Size constrained unequal probability sampling with a non-integer sum of inclusion probabilities. Electron J Stat 6:1477–1489

Hanif M, Brewer KRW (1980) Sampling with unequal probabilities without replacement: A review. Int Stat Rev 48:317–335

Horvitz DG, Thompson DJ (1952) A generalization of sampling without replacement from a finite universe. J Am Stat Assoc 47(260):663–685

Jauslin R, Panahbehagh B, Tillé Y (2022) Sequential spatially balanced sampling. Environmetrics 33(8):e2776

Jauslin R, Tillé Y (2020) Spatial spread sampling using weakly associated vectors. J Agric Biol Environ Stat 25(3):431–451

Madow WG (1949) On the theory of systematic sampling, II. Ann Math Stat 20:333–354

Narain RD (1951) On sampling without replacement with varying probabilities. J Indian Soc Agric Stat 3:169–174

R Core Team (2022) R: a language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria

Sunter AB (1977) List sequential sampling with equal or unequal probabilities without replacement. Appl Stat 26:261–268

Sunter AB (1986) Solutions to the problem of unequal probability sampling without replacement. Int Stat Rev 54:33–50

Tillé Y (1996) An elimination procedure of unequal probability sampling without replacement. Biometrika 83:238–241

Tillé Y (2006) Sampling Algorithms. Springer, New York

Tillé Y (2019) A general result for selecting balanced unequal probability samples from a stream. Inf Process Lett 152:1–6

Vijayan K (1968) An exact *πps* sampling scheme, generalization of a method of Hanurav. J Roy Stat Soc B30:556–566

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.