Using network flow optimization to sample with unequal probabilites

Miguel Sousa Lobo [*] Alan Gous [†]

August 5, 2002

Abstract

We present an efficient, exact algorithm for sampling k out of n items, without replacement, given the marginal probability that each item should be included in the sample. The algorithm optimizes network flow on a lattice constructed from the tree of conditional probabilities. The problem is motivated by an industrial application: the random selection of items to merchandise on a web page, subject to specification of each item's display frequency.

1 Introduction

We wish to draw a random sample of size k from n items, without replacement. For each item i of these n we are given the marginal probability that the item should appear in the sample.

A number of methods of doings this have been proposed in the literature, of varying degrees of complexity and applicability. In this paper we present a fast, exact, and generally applicable method, obtained by solving a network flow optimization problem on a lattice network.

To state the problem formally, let $I_1, \ldots, I_n$ be indicator random variables each taking value 0 or 1. For each i, if $I_i = 1$ the ith item is chosen, otherwise not. We wish to draw $I_1, \ldots, I_n$ from a joint probability distribution which satisfies

$$\mathrm{Prob}(I_i = 1) = \pi_i, \quad i = 1, \ldots, n, \tag{1}$$

and

$$\sum_{i=1}^{n} I_i = k. \tag{2}$$

These two conditions, together with linearity of expectation, imply

$$\sum_{i=1}^{n} \pi_i = k, \tag{3}$$

[*] mlobo@duke.edu

[†] alan@cariden.com

which we can assume is satisfied by the given π_i. Also, without loss of generality, we can assume that $0 < \pi_i < 1$ for all i.

There are many joint distributions that satisfy these conditions. A useful and interesting feature of the method presented here is that joint distributions with various favorable sampling properties may be obtained by varying the choice of objective of the flow optimization problem.

Sampling without replacement has found most application in survey sampling. Brewer and Hanif ([BH82]) list 50 methods of drawing these samples, and discuss their merits and demerits within the survey sampling context. Our work was motivated by a different application: the selection of merchandising items to display on a web page.

Section 2 discusses this application. Sections 3–5 develop the proposed method. Section 6 discusses sampling properties resulting from the use of certain optimization objectives, while comparing the method to some previously proposed.

2 An Application

The problem of sampling without replacement arises in the following industrial application. A few thousand products are to be merchandised on a web-page. Each page includes a number of merchandising boxes (say, three to five) where the products are displayed. Each time a user views the page on their browser, these boxes are to be filled with a selection from the complete set of products. The amount of exposure that each product should receive is regulated. Formally, *display frequencies* specify the proportion of page-views in which each product is to appear. Suppose that n products are available for selection, the web page includes k boxes, the display frequency of product i is π_i, and the indicator I_i of Section 1 defines whether product i should be displayed or not. To fulfill the requirements on the product exposures we therefore need an algorithm to perform the task described in Section 1.

An important goal in this sort of electronic merchandising is to collect information about the attractiveness of each product to users. One way this is done is by estimating the probability that a user will click on any product (or put the product in their shopping cart, or buy it), given that it was displayed in a merchandising box.

The procedure differs from standard survey sampling in a number of respects. Firstly, the goal is to estimate a property, say the probability of being clicked on, of each product in the population, rather than a property of the population as a whole. Also, a measurement depends on the entire sample displayed, rather than each sample element separately. Lastly, in the choice of products to display there are conflicting goals. There is a trade-off between gathering information through showing a wide variety of products, and achieving a high “click-through” rate by showing just the most popular products. This tradeoff can be studied in the context of Bandit Problems (see, e.g., Gittens [Git89]). We will not consider this issue here, but instead start our analysis with specified π_i, however they are chosen.

Because the sampling procedure plays a role in a form of survey, additional requirements on the procedure need to be enforced. The combinations of items that appear together on the same page should be, in some sense, randomized, since if an item always appears next to some other item that happens to be very popular, its attractiveness may be underestimated. Variances of estimators of the surveyed variables also depend on which combinations of items can appear together. We will return to these requirement in Section 6.

3 A Direct method

If we specify directly all n^k points in the joint probability, each sample request can be fulfilled using a single uniformly distributed continuous random variable. The obvious drawback is the storage requirement for the n^k probabilities.

The computation of the joint probabilities can, however, be framed as a linear feasibility problem. For example, for the case $k = 2$, define

$$p_{ij} = \mathrm{Prob}(I_i = 1,\, I_j = 1,\, I_k = 0 \text{ for } k \neq i, j), \quad i < j. \tag{4}$$

Then the set of valid joint probabilities is defined by the constraints

$$0 \leq p_{ij} \leq 1,\; i, j = 1, \ldots, n \tag{5}$$

$$\sum_i \sum_j p_{ij} = 1 \tag{6}$$

$$\sum_j p_{ij} + p_{ji} = \pi_i,\; i = 1, \ldots, n \tag{7}$$

$$p_{ij} = 0,\; i, j = 1, \ldots, n,\; i \geq j. \tag{8}$$

In mathematical programming terms, the $p \in \mathbf{R}^{n \times n}$ are the problem variables, and the $\pi \in \mathbf{R}^n$ are the problem data.

An objective function may be added to select a particular point in the feasible set. This objective can be selected based on the properties that are desirable for the joint distribution. For example, Raj ([Raj56], see Brewer and Hanif [BH82]), suggests a linear objective for the case $k = 2$ which minimize the variance of a specific sampling estimator.

For the application described above we wish to distribute the probability mass evenly, in some sense, over the points in the joint distribution, and objectives such as entropy maximization could be considered.

However with, say, $n = 2,000$ and $k = 5$, which are reasonable numbers for the application described above, n^k becomes an unmanageably large number.

The choice of objective will be discussed in detail in Section 6, for an optimization problem of a more practical size.

4 Constructing a Decision Tree

As an alternative, consider the following procedure. Sequential decisions are made on whether to select each item i, from 1 to n. This is represented by the decision tree in Figure 1.

Each node in the tree is associated with a conditional branch probability. This is the probability that item $i+1$ is selected, given the previous i selection decisions. Formally, we define

$$b(\, i+1 \,|\, s_1, \ldots, s_i \,) = \text{Prob}\,(\, I_{i+1} = 1 \mid I_1 = s_1, \, \ldots \, , I_i = s_i \,)\,. \tag{9}$$

Instead of solving for p as described in the previous section, we could consider solving directly for the b variables.

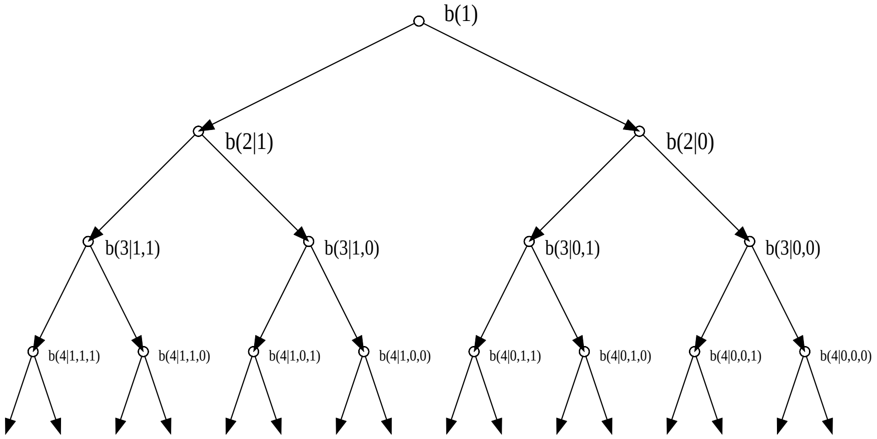

Figure 1: Decision tree for $n = 4$.

This procedure automatically enforces the no repetition condition, but is not practical. Firstly, the branch probabilities are difficult to handle as variables of a mathematical program. Specifically, the marginal constraints specified by the π_i are not linear in the branch probabilities. More importantly, the number of nodes grows with 2^n, so the pre-computation and storage of b is not practical.

The tree could be pruned based on the problem constraints, since after k positive decision branches have been followed, all branching probabilities in the child nodes must be zero, and after $n-k$ negative decision branches have been followed, all branching probabilities in the child nodes must be one. But this still results in a large and somewhat more complex structure.

5 Reduction to a Lattice

We now consider a variation on the above method in which the tree is collapsed into a lattice. This is achieved by imposing extra constraints on the branch probabilities.

Defining $s^i = (s_1, \ldots, s_i)$, we require that

$$b(\, i+1 \,|\, s^i \,) = b(\, i+1 \,|\, perm(s^i)\,), \tag{10}$$

for all permutations *perm* of s_i. Put another way, we are now conditioning on the probability of selecting each item based only on the number of items already selected, and not, as before, on which particular items were selected. Denoting the number of items already selected as

$$t_i = \sum_{j=1}^{i} s_j^i, \tag{11}$$

we can then rewrite these constrained branch probabilities simply as $b(i+1|t_i)$.

This procedure effectively "merges" a number of nodes in the tree. In Figure 1, for example, the pair of nodes associated with $b(3|0,1)$ and $b(3|1,0)$ merge, as does the triple $b(4|1,0,0)$, $b(4|0,1,0)$ and $b(4|0,0,1)$, and the triple $b(4|1,1,0)$, $b(4|1,0,1)$, and $b(4|0,1,1)$.

A tree with nodes merged in this way, and pruned as described at the end of the previous section, becomes a rectangular lattice of dimension $k+1$ by $n-k+1$. Such a lattice, for the case $k=2$ and $n=5$, is represented in Figure 2. The lattice has been rotated 45 degrees counter-clockwise compared to the illustrated tree. The role of the two extra branches, at the top left and bottom right, as well as the x and y variables, will now be explained.

To compute appropriate values for the $b(i+1|t_i)$, we change variables. We work with the probabilities that branches of the lattice are traversed in the course of drawing a sample. We define x_{ij} to be the probability that item $i+j-1$ is selected and, out of the previous $i+j-2$ items, $i-1$ were selected and $j-1$ were not. Likewise for y_{ij}, but for the probability that item $i+j-1$ is not selected. Formally,

$$\begin{aligned} x_{ij} &= \text{Prob}\left(I_{i+j-1}=1, t_{i+j-2}=i-1\right), & (12)\\ y_{ij} &= \text{Prob}\left(I_{i+j-1}=0, t_{i+j-2}=i-1\right). & (13)\end{aligned}$$

Here the i subscript takes values from 1 to k in the case of x, and to $k+1$ in the case of y. The j subscript takes values from 1 to $n-k+1$ in the case of x, and to $n-k$ in the case of y.

Under the appropriate constraints on x and y, there is a one-to-one map relating these variables to branching probabilities. The reverse mapping is

$$b(i+j-1|i-1) = \frac{x_{ij}}{x_{ij}+y_{ij}}. \tag{14}$$

This now becomes a network flow problem (as a general reference see, for example, Bertsekas [Ber98]). For the x and y to define a flow on the lattice that can be mapped to a set of branch probabilities, the following constraints must be met:

- (C1) *Non-negative flow*: $x_{ij} \geq 0, y_{ij} \geq 0$.
- (C2) *Flow conservation*: At each node, the sum of incoming probabilities must equal the sum of outgoing probabilities. So $x_{ij}+y_{ij} = x_{i-1,j}+y_{i,j-1}$ for all i and j, where the obvious adjustment should be made for boundary conditions: omit terms with indices out of bounds.

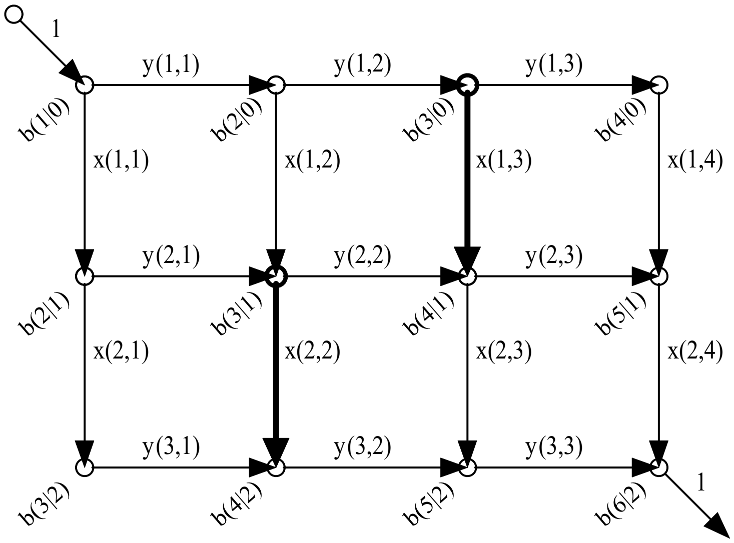

Figure 2: Decision lattice to select 2 out of 5.

- (C3) *Input flow*: $x_{1,1} + y_{1,1} = 1$.
- (C4) *Output flow*: $x_{k,n-k+1} + y_{k+1,n-k} = 1$.

To make the flow in Figure 2 more intuitive, two edges have been added, carrying a total flow of 1 into and out of the network.

The constraints on the marginals are readily expressed in terms of the new variables. The sum of alternate branches (the x branches) in each level must equal π_i. Note that the levels of the tree or lattice correspond to anti-diagonals when the lattice is rotated +45 degrees as in Figure 2, where, as an example, the branches pertaining to q_3 are drawn in bold. More formally,

$$\sum_{i=1}^{k} x_{i,j-i+1} = \pi_j,\ j = 1, \ldots, n, \tag{15}$$

where we define $x_{ij} = 0$ when the index is out of bounds.

Since all the above constraints are linear in the branch traversal probabilities, we again have a linear feasibility problem. The number of variables is of order nk, which is readily handled for the values typical in the application discussed above. A proof of feasibility for every n, k and π_i satisfying (3) is provided in Appendix A.

It is clear that any sampling procedure which proceeds sequentially through the population vector, and for which the probability of selection of each population element depends only on the number of elements selected before it, can be represented as a solution to this feasibility problem. Chromy's sampling procedure (Brewer and Hanif, [BH82]), is one such implicit solution. This sequential procedure ensures at each step i from 1 to n, the expected number of sample elements already selected is always $\sum_{j=1}^{i-1} \pi_j$.

Specific feasible solutions can also be obtained through the specification of an objective function. Some possibile objectives will be considered in Section 6.

To summarize the method, we have two stages:

1. *Lattice design stage:* Solve the feasibility problem defined by constraints (C1) through (C4) and (15). (Or, alternatively, solve an optimization problem, adding a suitable objective function.)

2. *Sampling stage:* Traverse the lattice, branching at each node according to the probabilities given by (14). Each alternative path through the lattice corresponds to a different sample.

This method has been successfully used in the electronic merchandising application described in Section 2. In the implementation, a random permutation was performed on each resulting sample to ensure that each item had the same probability of appearing in each box on the web page.

In a typical implementation of this application, multiple servers and processes would serve web-pages in parallel. An applet is created for each user session. Because of this, a stateless algorithm such as the one just described is preferred, to avoid the need to implement locks on the algorithm data, which would slow down the overall system, and create scalability issues.

Computation time is a concern in such an application, since it adds to the overall delay in serving the web-page. Since the sampling procedure is repeated a large number of times using the same solution to the mathematical program, Step 1 of the procedure above is performed offline. The much shorter Step 2 is then performed each time a web-page is requested.

6 Objectives

From among the solutions to the feasibility problem of Section 5, some may be preferable to others. In this section, we briefly discuss some reasons for such preferences. We also propose objective functions that address those preferences and that, when coupled with the constraints, result in an optimization problem that can be solved efficiently.

The pair-wise probabilities

$$\pi_{ij} = \mathrm{Prob}\,(I_i = 1, I_j = 1) \tag{16}$$

play an important role in the quality of a sampling method. It has already been mentioned in Section 2 that a goal of the web-merchandising sampling was to show many different combinations of products at the same time, both for variation and to help in estimating relatively more popular products. This can be achieved through sampling methods with large, positive π_{ij}. In the classical theory of survey sampling without replacement (see Brewer and Hanif, [BH82]), the standard variance estimator, the Yates-Grundy estimator, relies on positive π_{ij} for unbiasedness, and large π_{ij} for stability.

The following objective of the lattice optimization ensures that all π_{ij} (and in fact all higher order marginals too) are non-zero:

$$\text{maximize } \min_{i,j} \{x_{ij}\} \cup \{y_{ij}\}. \tag{17}$$

The optimization ensures that all $x_{ij} > 0$ and all $y_{ij} > 0$, which ensures in turn that no branching probabilities (14) are 0 or 1, so that all paths through the lattice can be traversed with non-zero probability, and therefore that any subset of size k of the population can be chosen with non-zero probability, from which the result follows.

The linear feasibility problem together with the above objective can be expressed as an LP, for which very efficient, polynomial-time algorithms exist.

We can change (17) to an objective which not only ensures that all k-subsets are possible samples, but also that the items selected are as close to independent as possible, in the sense that the branching probabilities (14) are close to their respective π_i. That is, we obtain solutions as close to

$$\frac{x_{ij}}{x_{ij} + y_{ij}} = \pi_{i+j-1}, \tag{18}$$

for all i and j, as possible.

We could use the quadratic objective

$$\text{minimize } \sum_{ij} \left(\frac{x_{ij}}{\pi_{i+j-1}} - x_{ij} + y_{ij} \right)^2, \tag{19}$$

which leads to a QP. However, this may yield branching probabilities of 0 or 1. Better is the convex objective

$$\text{minimize } \sum_{ij} \log(x_{ij}) + \frac{\log(y_{ij} - x_{ij})}{\pi_i + j - 1}. \tag{20}$$

The minimum of each term in this objective is, like that of (19), obtained at (18). However, the terms approach infinity as the branching probabilities approach 0 or 1, so ensuring that all k-subsets have non-zero probability of occurring.

Note that the convex optimization problem resulting from the use of the above objective can be solved easily using some variation of Newton's method, without the need for a barrier function for the inequality constraints, since the objective is approaches infinity at the boundaries of the feasible region.

7 Notes

For furhter details on the implementation of the lattice method, including Matlab code, see `http://faculty.fuqua.duke.edu/~mlobo`.

The authors would like to thank Arman Maghbouleh, Eunhei "PJ" Jang, David Brown, and Arash Afrakhteh for help in the implementation of the method, and Art Owen for helpful comments on a draft of this paper.

A Proof of General Applicability

Write the problem constraints as

$$A^T z = 0 \tag{21}$$
$$a_1^T z = 1 \tag{22}$$
$$a_2^T z = 1 \tag{23}$$
$$z \geq 0 \tag{24}$$
$$P^T z = \pi, \tag{25}$$

where $z = \left[x^T y^T\right]^T \in \mathbf{R}^m$. Constraint (21) is the flow conservation, and constraints (22) and (23) are the source and sink flows.

Choose a path through the lattice from source to sink, and let $t \in \mathbf{R}^m$ be a vector such that $t_i = 1$ on the path, $t_i = 0$ otherwise. Note that $z = t$ satifies constraints (21) through (24).

Let $T \in \mathbf{R}^{m \times \binom{n}{k}}$ be a matrix with a column associated in this manner with each possible path through the lattice. For any $\lambda \in \mathbf{R}^{\binom{n}{k}}$ such that $\lambda \geq 0$ and $\mathbf{1}^T \lambda = 1$, and by linearity, $z = T\lambda$ also satisfies constraints (21) through (24). We show there is a λ for which, in addition, $P^T T \lambda = \pi$.

Note that the path must traverse exactly k vertical edges, and that each anti-diagonal is only traversed once. Hence, $P^T t$ has exacly k entries equal to one (the others being zero). Define $Q = P^T T \in \mathbf{R}^{n \times \binom{n}{k}}$. The columns of Q are all the possible sets of k non-zero entries.

By Farkas' lemma (see, e.g., [BT97, §4.6]), if there is no λ such that

$$Q\lambda = \pi \tag{26}$$
$$\lambda \geq 0, \tag{27}$$

there must be a v such that

$$v^T Q \leq 0 \tag{28}$$
$$\pi^T v > 0. \tag{29}$$

Suppose we have such a v. Label the entries of v so that

$$v_{[1]} \geq v_{[2]} \geq \cdots \geq v_{[n]}. \tag{30}$$

Given the structure of Q, the condition $v^T Q \leq 0$ is equivalent to

$$\sum_{i=1}^{k} v_{[i]} \leq 0. \tag{31}$$

But, from $0 \leq \pi \leq 1$ and $\mathbf{1}^T \pi = k$, we have that

$$\pi^T v \leq \sum_{i=1}^{k} v_{[i]} \leq 0, \tag{32}$$

which shows there is no such v and completes the proof.

References

[Ber98] D. Bertsekas. *Network Optimization.* Athena Scientific, 1998.

[BH82] K.R.W. Brewer and M Hanif. *Sampling with Unequal Probabilities.* Springer-Verlag, 1982.

[BT97] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1997.

[Git89] J.C. Gittins. *Multi-armed bandit allocation indices.* Wiley, New York, 1989.

[Mad49] W.G. Madow. On the theory of systematic sampling, ii. *Annals of Mathematical Statistics*, 20:333–354, 1949.

[Raj56] D. Raj. A note on the determination of optimum probabilities in sampling without replacement. *Sankhya*, 17:197–200, 1956.