Computer Assisted Probability Sampling in the Field

Pushpal Mukhopadhyay and Sarah Nusser
Department of Statistics
Iowa State University
Ames, Iowa, 50011

Abstract:

Until recently, probability sampling designs for field studies were largely limited by the difficulty of locating random points in the field. The ability to receive precise GPS signals in many field settings has largely removed this constraint. We have been investigating statistical and computer-based technologies that enable field data collectors to use more sophisticated probability sampling designs. In particular, our research has focused on a scenario in which a field data gatherer can use a mobile computer to download geospatial data for the target region from the Internet and use the geospatial data layer as a basis for selecting a probability sample real-time in the field. We will discuss this research through an example with prototype software. The software allows stratified random sampling with geographic strata defined as regular rectangles on an arbitrary geographic image covering the target area, of as classes derived from a discrete-valued image such as a land cover map. The software also could be extended for adaptive sampling designs for detecting rare events. We will discuss extensions of this research that include using covariate data layers to determine selection probabilities.

1. Introduction

Federal statistical agencies and researchers collect critical data on the nation's natural resources. These data are sometimes gathered largely by mobile field data collection. But because of the difficulty of locating sample points in the field, probability sampling for field studies was often constrained to simple designs in the past. The emergence of commercially available GPS receivers makes it possible for the data gatherers to navigate to arbitrary sample locations in the field.

Our objective is to improve the ability of data gatherers to use geospatial information in the field. In this direction, we propose a field software environment that will enable data collectors to implement different probability sampling designs for field surveys.

We have developed prototype software that will implement different sampling strategies for field data collection. The software allows user to download geospatial data for the target region from the Internet. The user can choose a sampling design such as simple random sampling, stratified sampling with geographic region defined as strata, stratified sampling with thematic information defined as a strata or adaptive sampling. The software will select the location of the sample points based on the sampling design. Data gatherers can then navigate to the specified sampling point in the field using GPS receiver and collect data. We will discuss the implementation of the software using a land cover image from Northeast Iowa. We also consider sampling schemes that utilize auxiliary information in selection probabilities. Finally, we are developing computer infrastructure model that will make it possible for the user to find, download and upload geospatial information from the Internet.

Section 2 describes several sampling schemes and how they have been implemented in the software. Section 3 extends this idea with an example of a land cover image. Section 4 presents some discussion and further questions for future work on this topic.

2. Sampling Designs

2.1 Simple Random Sampling

A sampling procedure that assures that each sample of n units from the population (image) has an equal chance of being selected is referred to as simple random sampling (SRS). One consequence of this design is that each element has an equal probability of being selected in the sample.

Estimation:
Let there are N pixels in the image and let $y = (y_1, y_2, ..., y_N)$ be the values for the variable of

204 Snedecor Hall, Ames, IA
pushpal@iastate.edu

interest associated with each pixel (for example say the pH of soil or the elevation of land, etc.). Suppose a simple random sample $s = (1, 2, ..., n)$ of size n has been observed from the above population. Then the mean for the entire population is $\bar{y} = \frac{\sum_{i \in s} y_i}{n}$, and an estimate of variance of the above population is $S^2_{ys} = \frac{\sum_{i \in s} (y_i - \bar{y})^2}{n-1}$
Implementation:

An image is a collection of pixels, and each pixel contains a color code. The color code is represented by the intensities of red, green, and blue values and is usually denoted by a triplet (R,G,B) of integers. Without loss of generality, for our purpose, we can assume that the image is stored in a computer as an array of pixels (raster data), i.e., an array of (R,G,B) values. The coordinate of the left top corner of the image is denoted by (0,0) and the x coordinates increase along the width and the y coordinates increase along the length (vertically downwards) on the picture. This coordinate system is known as picture coordinate system.

A land cover map is an image of the above type associated with a world file. The world file contains the following information:
1. World coordinate (latitude, longitude) of the 1^{st} pixel at the top left corner. For example, the world coordinate of the pixel associated with the (0,0) picture coordinate, might be (w_x, w_y).
2. Extent of each pixel along the X direction, say l_x.
3. Extent of each pixel along the Y direction, say l_y.
4. Tilt of the land with X axis of the picture coordinate, say θ, the inclination of the picture coordinate with the world coordinate.

Hence the world coordinate of any pixel whose picture coordinate is (x, y) is given by $(w_x + xcos\theta - ysin\theta, w_y + xsin\theta + ycos\theta)$ and the area of each pixel is $l_x \times l_y$ on the land. So, without loss of generality, we can work with picture coordinates and can show the world coordinate of each sampled pixel so that user can navigate to the sampling location using GPS device.

To implement SRS, we first find out the dimension (length and width of the image in pixels) of the downloaded image. Then for each sample point we choose two independent random numbers (X and Y picture coordinates) within that range. Each sample point is chosen using the random number generator with different seeds. Sampling is done with replacement of chosen points. The sample mean and standard error for the mean statistics are provided dynamically.

There are a number of potential problems with simple random sampling. The collected sample may not be representative of the population. We have no control over the subpopulation, and precision can be increased through other types of sampling.

2.2 Stratified Sampling

In this random sampling technique, the whole population is first divided into mutually exclusive subgroups or strata, and then units are selected randomly from each stratum. The strata are based on some predetermined criteria such as geographic location, area, or demographic characteristic. We have implemented two different types of stratification, namely, positional or geographic stratification and thematic stratification. In positional stratification, we divide the downloaded land cover image into small rectangular strata of equal area. The user can define the number of rows, number of columns, and number of sample points for each stratum. In thematic stratification, we divide the entire image based on the land cover type (thematic value of each pixel), and user can specify the number of sample points from each land cover.

Estimation:
Let there are N pixels in the image and let $y = (y_1, y_2, ..., y_N)$ be the values for the variable of interest associated with each pixel (for example, the pH of soil, the elevation of land etc.). Now let us assume that the N units of the population U can be divided as $U = (1, 2, ..., N) = \cup_{h=1}^{H} U_h$ where $U_h \cap U_g = \emptyset$, for $h \neq g$ and, let N_h be the size of the h^{th} strata. So for a simple random sample of size n_h within stratum h, the population mean is estimated by $\bar{y}_U = \sum_h \frac{N_h \bar{y}_{U_h}}{N}$, where $\bar{y}_{U_h} = \frac{\sum_{k \in S_h} y_k}{n_h}$ is the estimation of mean for each strata. Also, the population variance within each strata is estimated by $S^2_{ys_h} = \frac{\sum_{s_h} (y_{hk} - \bar{y}_{s_h})^2}{n_h - 1}$, and $var(\bar{y}_U)$ is estimated by $\sum_h (\frac{n_h}{N_h})^2 \frac{\sum_{s_h} (y_{hk} - \bar{y}_{s_h})^2}{n_h - 1}$

Implementation:
a) Stratified sampling with positional strata: Suppose the user wants to divide the entire image into rectangular cells (strata) arranged in c columns and r rows. Find the length and width of the image and then divide the width by c and length

by r to get the dimension of each strata, say w and l respectively. The starting coordinate for the stratum which is on the i^{th} row and j^{th} column is given by $((i-1)w, (j-1)l)$. Now generate a random number, say a, from 0 to w to get the x coordinate of the sample point and a random number, say b, (independent of other random numbers) from 0 to l to get the y coordinate. If (x_h, y_h) is the starting coordinate of the h^{th} strata, then $(x_h + a, y_h + b)$ is the selected sample point. Repeat the process until all the sample points are chosen. We have implemented an allocation with an equal number of sample points within each stratum.

b) Stratified sampling with thematic strata: In this type of stratification, the user wants to select different numbers of sample points from different land covers or color codes. Using the values of the 'color code', define the strata in this situation. We have implemented STRS with thematic stratification using a rejection algorithm approach. First scan the entire image and find out how many different color codes are there. Now choose a random point (x_i, y_i) from the entire image and find its color code. If the chosen point belongs to a stratum (determined by the color code), and if that stratum needs more points, then include that chosen point as a sample point for that stratum; otherwise reject the chosen point.

Another approach is to scan the entire image and put the different color codes in different arrays. Each array will thus contain picture coordinates of pixels for that particular color code. For example, if we have 10 different colors in the image, then we will have 10 arrays containing the picture coordinates of each color. If the length of the array for one particular color code (say h) is N_h and we need n_h sample points from that strata, then we choose n_h random numbers independently from $(1, 2, ..., N_h)$, which will give the array index for the sample points. The sample points are then given by the coordinates of indices in the sample array.

Among these two approaches, the rejection algorithm approach takes less space, but the second approach is faster. If we have a color in the image that is 10% or less of the population, then the second approach improves performance substantially. Yet considering the limited computing ability of many field computers, the rejection algorithm approach may be preferred over the second, since, it takes much less memory.

2.3 Sampling using map accuracy information

The downloaded image (say, a land cover map) may not be 100% accurate. There are different sources of variation associated with the map-making process. Positional accuracy (change of the relative position of a pixel in the map) and misclassification of a pixel (classified into a different land cover) are the most common sources. In this random sampling approach, we will use an estimated accuracy measure as a size measure to draw our sample.

Available accuracy information: Suppose we have some measurement accuracy associated with each pixel in the map. We can think this as covariate information and use this information to improve our design-based estimation. Let $U = (1, 2, ..., N)$ be the total population and $U = \cup_h U_h$ with $U_h \cap U_k = \Phi$ for all h and k, where U_h denotes the index set for a particular color code. Let us assume that the probability of i^{th} pixel in the map belongs to the h^{th} domain in the field is known (from an accuracy study). Now we will use this probability information to improve our estimate. For example, let us assume that we can model, $logit(p_{ih}) = X^t\beta$, where p_{ih} denotes the probability that i^{th} pixel on the map belongs to the h^{th} strata in the field, X_{1i}= richness for i^{th} pixel (i.e. number of unique land cover in 3×3 neighborhood of i^th pixel), X_{2i}= number differences (i.e. number of pixels in the 3×3 neighborhood which have a different land cover than the i^{th} pixel). The estimate of β is obtained from the accuracy study.

We want to choose a sample of size n_h from stratum (land cover) U_h based on the auxiliary information given by the respective probabilities $\vec{p_h} = (p_{1h}, p_{2h}, ..., p_{Nh})$. We will use probability proportional to size sampling, where the accuracy measure for each pixel will be considered as a size measure. In this way we will tend to choose points with high p_{ih}.

Estimation: Let there be N pixels and let $y = (y_1, y_2, ..., y_N)$ be the values for the variable of interest associated with each pixel. Suppose that N units of the population U can be divided as $U = (1, 2, ..., N) = \cup_{h=1}^{H} U_h$, where $U_h \cap U_g = \emptyset$, for all $h \neq g$. Moreover, let us assume $p_{ih}, i = 1, 2, ..., N$ are the probabilities that i^{th} pixel belongs to the h^{th} domain and are known for all i and h. With this set-up, a design unbiased estimate of total is $\hat{T}_\pi = \sum_k y_{i \in S}/p_{ih}$ and an estimate of mean of y on U_h is $\frac{\hat{T}_\pi}{N_h}$, where N_h is the size of U_h.

To estimate the variance of $\hat{T}_\pi$ let $t_{kh} = \sum_{i=k}^{N} p_{ih}$, $k^* = min\{k_0, N-n+1\}$, where $k_0 = inf\{k : \frac{np_{kh}}{t_k} \geq 1\}$ and $T_N = \sum_{iinU} p_{ih}$ and let $g_k = g_{k-1}\frac{t_{kh}-p_{k-1h}}{t_{k+1h}}, g_1 = \frac{1}{t_{2h}}$, for all $k = 2, 3, ..., k^* - 1$.
Then for U_h, it can be shown that the simultaneous inclusion probability of k^{th} and l^{th} pixel is,

$$\pi_{kl,h} = \begin{cases} \frac{n(n-1)}{T_N} g_k p_{kh} p_{lh} & ,1 \leq k < l < k^* \\ \frac{n(n-1)}{T_N} g_k p_{kh} \bar{p}_{k^*h} & ,1 \leq k < k^* \leq l \leq N \\ \frac{n(n-1)}{T_N} g_{k^*-1} \frac{t_{k^*h}-p_{k^*-1h}}{t_{k^*}-\bar{p}_{k^*h}} (\bar{p}_{k^*h})^2 & ,k^* \leq k < l \leq N \end{cases}$$

Hence a variance estimator of $\hat{T}_\pi$ is given by,
$\hat{var}(\hat{T}_\pi) = \sum_{k\in S} \sum_{l\in S} (\pi_{kl,h} - p_{kh}p_{lh})(\frac{y_k}{p_{kh}} \frac{y_l}{p_{lh}})$

Implementation: Suppose user wants to choose n_h sample point from U_h strata. To implement this sampling scheme for the h^{th} strata, we first generate a random number from 0 to N, say i. Now we generate another random number from $U(0, 1)$ distribution, say u. If $p_{ih} > u$ then include that point into the sample; otherwise reject the i^{th} pixel. When p_{ih} for all pixels are available (i.e. we do not need to calculate it dynamically using the function for p_{ih}), we can use a systematic probability proportional to size design to choose our sample points.

Among these two methods, the first method is useful if we have minimal computing capabilities because in this method we don't have to calculate p_{ih} for all the pixels in the image. (A county of size 20 mi × 30 mi has approximately 2 million pixels.). But in the second method, we need only one draw to specify all sample points. So if p_{ih}'s are provided with the map, then this approach is more useful.

3. Examples

We developed prototype software and implemented all the above-mentioned sampling methods. To develop the prototype software we also considered the limited computing capabilities of field computers. We developed the software using JAVA, and it has small screen interface capabilities.

In this section we will discuss how our prototype field sampling software works with an example. The downloaded map is a land cover map TIFF image for a four-county region in northeast Iowa. This data served as our remotely-sensed land cover map, with the smallest unit of map, or pixel, corresponding to a 30 meter × 30 meter area on the ground. Pixels into one of 17 classes based on their 'Color Code'. The following table shows the land cover types and their color codes.

Land cover type	RGB values
Open Water	(0,2,254)
Low Intensity Residential	(200,200,200)
High Intensity Residential	(175,175,175)
Commercial, Industrial	(152,152,152)
Bare Rocks, Sand, Clay	(178,101,81)
Quarries, Strip Mines	(189,15,16)
Transitional	(255,197,115)
Deciduous Forest	(5,133,0)
Evergreen Forest	(9,119,6)
Mixed Forest	(206,211,109)
Grassland	(116,213,111)
Pasture, Hay	(187,235,125)
Row Crops	(252,255,164)
Small Grains	(254,255,75)
Urban	(189,189,189)
Woody Wetlands	(87,144,112)
Herbaceous Woodlands	(15,169,233)

Suppose a data gatherer (DG) wants to implement a sampling scheme with 12 sample points from Open Water, 25 sample points from Deciduous Forest and 50 sample points from Pasture. Then the DG will collect data for the elevation and the pH of the soil.

First the DG will use the query screen to download the Iowa land cover image into the DG's computer. Then the menu option '$Sampling- > StratifiedSampling- > ThematicStrata$' will scan the entire image and present a list of all available land covers with their size (in number of pixels) for the downloaded image.

Now the DG can choose the required number of sample points for each land cover. The software will then choose the sample points based on this sampling design and will show the chosen sample points by little red plus signs on the image. If the user taps on the selected point, then it will show the picture coordinate and the world coordinate of that point. By using the GPS signal, a data gatherer can physically go to that location (identified by the world coordinate) and collect the information on elevation and pH of the soil for that point. After the DG collects data, the software will provide design unbiased statistics of the mean and standard error of elevation and pH of the soil on request.

Similarly, to do positional stratification, the data gatherer will provide the number of rows and columns in which the DG wants the area to be divided. And after the DG choose the number of

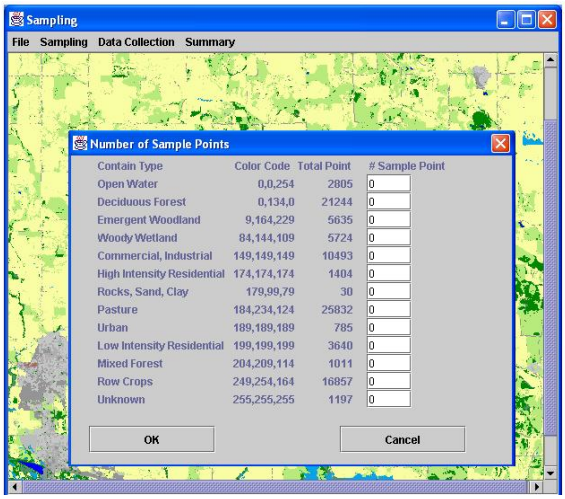

Figure 1: Land Cover Map

Figure 1 shows the different land covers available in the downloaded map with their color codes and total number of pixels for each land cover in the map. Through the last column user can input the number of required sample points for that land cover type.

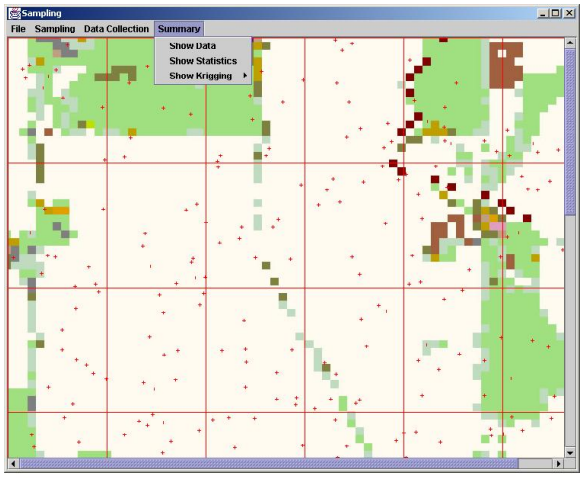

Figure 2: Positional Strata with selected sample points

Figure 2 shows the screen shot after doing a positional stratification and choosing 10 sample points from each strata. Using the summary menu option shows user can get summary statistics.

required points from each stratum, the software will select sample points for the DG. Now the DG can go to the sample location in the field using the geographic coordinate shown in the software and GPS and collect data. After collecting data, the DG can obtain the mean and standard error for each variable.

For simple random sampling (a special case of positional stratification with one row and one column), the data gatherer needs to specify total number of sample points the DG wants. The software will choose the sample points for the DG and will show the DG the geographic location for each selected point. So the DG can navigate to the sampling location using GPS and collect data, the software will also provide the design unbiased estimator for each defined variable.

For sampling using accuracy information, we first produce the accuracy surface graphically. For each pixel we draw the p_{ih} values for a particular color code h by different intensities of red. Where a red pixel (RGB=(255,0,0)) implies $0.8 \leq p_{ih} \leq 1$ and a white pixel (RGB=(255,255,255)) implies $0 \leq p_{ih} < 0.2$ and so on. Then user can choose number of sample points for that particular color.

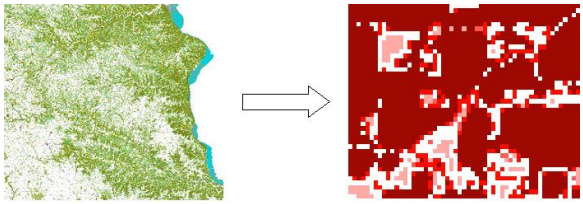

Figure 3: Map Accuracy Surface

Figure 3 shows the accuracy surface for evergreen forest. The figure on the left is the original downloaded map and the figure on the right is the accuracy surface for a portion of it. The red color shows a high probability of evergreen forest and white pixels shows a low probability.

4. Discussion

In this paper we discuss some basic sampling strategies and how they have been implemented in the computer. We use a digital image as our

sampling frame and a pixel as our sampling unit.

All of the above sampling approaches are appropriate for bmp, jpg, jpeg, gif, png or tiff types of encoding, except those where the pixel values are the basis of selection (such as stratified sampling with thematic strata and sampling with map accuracy information; The compression of the image does matter.) When the pixel value is the basis of selection, we should use a "lossless" compression as compared to a "lossy" compression. A lossless compression retains all information on color codes. It looks for more efficient ways to represent an image, while making no compromise in accuracy. In contrast, lossy compression accepts some degradation in the image to achieve smaller file size. Using lossy compression for sampling schemes when pixel values are not the basis of selection we can save space and computing time.

In the future, we will consider additional sampling designs, such as adaptive sampling to detect rare species or contamination hot spots within some specified region. Apart from design based estimators, we will also consider model based and model assisted estimators when covariate information for each pixel is available.

References

[1] C. Sarndal, B. Swensson, J. Wretman. *Model Assisted Survey Sampling*. Springer, 1992

[2] A.B.Sunter. *List sequential sampling with equal or unequal probabilities without replacement*. Applied Statistics **26**, 261-268, 1977

[3] Cay. S. Horstmann, Gary Cornell *Core Java, Vol. 1-2*. Sun Microsystem Press, 1999