GOVERNMENTS DIVISION REPORT SERIES

(Research Report #2010-4)

Stratification of a Sampling Frame with Auxiliary Data into Piecewise Linear Segments by Means of a Genetic Algorithm

Joseph J. Barth
Yang Cheng

U.S. Census Bureau
Washington, DC 20233

CITATION: Barth, Joseph J., Yang Cheng. 2010. Stratification of a Sampling Frame with Auxiliary Data into Piecewise Linear Segments by Means of a Genetic Algorithm. Governments Division Report Series, Research Report #2010-4

Report Completed: September 23, 2010
Report Issued: October 13, 2010

Disclaimer: This report is released to inform interested parties of research and to encourage discussion of work in progress. The views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

Stratification of a Sampling Frame with Auxiliary Data into Piecewise Linear Segments by Means of a Genetic Algorithm[1]

Joseph J. Barth[1], Yang Cheng[1]

[1]U.S. Census Bureau, Suitland, MD

Abstract

Regression-based estimators are used for several surveys conducted by the Governments Division of the U.S. Census Bureau. One assumption of this style of estimator is that the auxiliary variable has a roughly linear relationship with the estimated variables. Estimation procedures for 2009 through 2013 of finance and employment surveys will take this into consideration, but the original stratifications of the samples still do not. In this paper we propose using concepts from genetic algorithms to stratify a frame with auxiliary data such that the data fit a linear relationship in each stratum. To do this, we make the assumption that we know the maximum number of partitions for a piecewise linear fit. In particular, we consider the case of dividing an interval into at most three piecewise linear segments, with concepts that generalize naturally to any arbitrary maximum. One goal is to determine a minimal spanning set of the interval to partition. We first explore two families of piecewise linear functions with different levels of perturbation. We then consider this method on simulated data that does not fit the model well. Finally, we consider the method on actual data. The method is applied with a fixed number of iterations.

Keywords: Piecewise linear, genetic algorithm, survey, stratification.

1. Introduction

The Annual Survey of Public Employment and Payroll (ASPEP) is an annual survey of state and local governmental units and is conducted in the years between censuses of governments. The survey provides estimates of employment and payroll for each state. Our goal is to be able to provide estimates at more detailed levels of aggregation while maintaining our current levels of quality and timeliness.

A stratified probability proportional to size (PPS) sample design is currently being used for the ASPEP. Strata are initially defined as state-by-type of government. Most strata are sampled via PPS and these strata are currently sub-stratified into small and large unit sub-strata using the Cumulative Square Root of the Frequency method (CSRF). This sub-stratification based on size allows us to implement a cut-off based sample design where we significantly reduce the sampling fraction in the small unit sub-strata.

[1] This report is released to inform interested parties of research and to encourage discussion of work in progress. Any views expressed on statistical, methodological, or operational issues are those of the authors and not necessarily those of the U.S. Census Bureau.

We consider two ways in which our stratification methodology can improve our estimates: one based on the size of a unit and another based on our estimation methodology. For both approaches we first order the units in each stratum in ascending order. When considering size, we then plot the cumulative percent of units summed versus the cumulative percent of data summed. We then fit a piecewise linear function to this curve, and the end points of the linear pieces define our sub-strata. When considering the estimator, we plot the regressor versus the regressand and fit a piecewise linear function to this curve to derive our sub-strata.

2. Background

The ASPEP is an annual survey of all state and local governments in the 50 states, plus Washington, D.C. The universe and frame are the same as those used in the Census of Governments, with updates made to reflect any births, deaths, or mergers that may have occurred. A unit is determined to be a government if it exists as an organized entity, has governmental character (such as the power to levy taxes), and displays substantial autonomy (i.e., considerable fiscal and administrative independence).

The ASPEP collects data on five variables, and derives two additional variables from these. The five variables collected are full-time employees, full-time pay, part-time employees, part-time pay, and hours worked by part-time employees. The first derived variable is total pay, which is simply the sum of full-time and part-time pay. The second is full-time equivalent, which is calculated by dividing the number of part-time hours worked by the standard number of hours in a workweek for full-time employees in the particular government, added to the number of full-time employees in that government.

A new sample is selected two years after each census. The most recent sample prior to the 2009 sample was taken in 2004. The samples for 2005, 2006, and 2008 were the same as the sample taken in 2004, with the addition of any births that may have occurred since the 2004 sample was selected.

The data for each unit are subdivided into twenty-three different functions, such as fire protection, sewerage, and hospitals. Not every unit has all twenty-three functions. For instance special districts and school districts typically only have one or two functions.

Data are published at national and state levels for state-only, local-only, and state and local aggregates. For example, we can view just state government data for Alabama or all state government agencies combined with all local governments in Alabama. We can view a national total for all state governments combined, or we can view a national total for all local governments combined. If we do not consider data from Washington, D.C. (which, although not a state is in large part treated as one for processing and analysis) this gives us 150 state-level estimate tables, and 3 national-level estimate tables.

3. Goals

Our first goal is to identify improvements to our cut-off methodology. Currently[1] we use the Cumulative Square Root of the Frequency method[2] to define size categories in our sample design. We would like to consider several other potential methodologies for size-based stratification before we formally begin the 2014 sample redesign cycle.

Our second goal is to bring our stratification more in line with the regression models used in our estimation design. Our estimation assumes that there is a linear relationship between a unit's data in the prior census and the current sample year. By identifying sub-strata where this relationship is true and persists over time, we hope to improve the quality of our estimates.

4. Methodology

For both challenges, stratification with respect to our regression model and stratification with respect to unit size, we use almost identical methodologies. We will first present the common methodology used. We will then detail the changes made to the methodology in order to address differences between the two problems.

For both cases the goal is to fit a piecewise linear function to an appropriately chosen data set scaled to fit the unit square [0,1]x[0,1]. In this paper we choose to divide the function into at most three linear pieces both to simplify the presentation as well as to avoid excessive stratification; even without our proposed stratification methodology some strata are too small to provide reliable direct estimates. Strata are then defined as the end points of the partition of [0,1] determined by the piecewise linear fitting, scaled back to the original data range.

Describing this common methodology in greater detail, the first step is to scale the data into the unit square by simply dividing every data pair (x_i,y_i) by the maximum x and y values to obtain $(x_i /x_{max},y_i/y_{max}) \in [0,1]x[0,1]$.

We then use an algorithm designed using concepts from genetic algorithms, an overview of which can be found in Langdon and Poli, 2002[3]. We alter the commonly used order of operations (selection, recombination, and mutation) in this paper and we instead first perform recombination, then mutation, then selection.

His paper now provides a brief description of the algorithm, followed by a more detailed and accurate description. Our algorithm begins with a set of possible partitions of the interval [0,1]. We then randomly generate a second set of partitions based off of this initial set. We calculate the Mean Square Error (MSE) of the linear regression models defined by each partition. A new set, of the same size as the original, containing the partitions whose models have the smallest MSEs is then created. We then return to our random generation step and continue the process until a fixed number of iterations is met.

To begin our algorithm we need to create an initial or the 0th population. To do this we pick a few (in our case, four, but any number could be chosen that is at least equal to the

number of end points of the desired partitioning of [0,1] which we obtain from the algorithm) equidistant points in [0,1]. We choose {0,1/3,2/3,1} to populate our first generation. To do this we create $m_{i0} = (0, x_{i01}, x_{i02},1)$ with $(x_{i01}, x_{i02}) \in \{ \{0,1/3,2/3,1\} \times \{0,1/3,2/3,1\} \mid 0 \le x_{i01} < x_{i02} \le 1 \}$. This gives us an initial population of six elements and every subsequent population will have six elements as well. This method gives us population members which cover a large range of possible population members without carrying a large number of members through each iteration.

The ith member of the jth generation will be named $m_{ij} = (0, x_{ij1}, x_{ij2},1)$ where the set $\{0, x_{ij1}, x_{ij2}, 1\}$ are the (ordered) end points of a partition of [0,1]. We will interpret the case where two or more end points are equal as a partition with only 2 or 1 segments. We allow $1 \le i \le n_g$, the number of members in each generation, and x_{ij1}, x_{ij1} in [0,1] with $x_{ij1} \le x_{ij2}$, $1 \le j \le$ maxIterations, where maxIterations is the number of iterations which have been selected for the algorithm to run.

Our recombination step is as follows: given m_{ij}, $1 \le i \le n_g$, we want to construct $m_{i(j+1)}$ $1 \le i \le n_g$. To this end we first construct mij1 $1 \le i \le n_g$, which will be a recombination of the m_{ij}'s. Let r_{ij1}, $r_{ij2} \sim$ unif($\{1,2,\ldots,n_g\}$). This selects two members of the jth generation, $m_{(r_{ij1})j}$ and $m_{(r_{ij2})j}$ which we will use as the parents of m_{ij1}. Next select r_{ij3},$r_{ij4} \sim$ unif($\{1,2\}$). We finally have that $m_{ij1} = (0, x_{(r_{ij_{rij3}})j1}, x_{(r_{ij_{rij4}})j2}, 1) = (0, x_{ij11}, x_{ij21}, 1)$. Without the notation, this simply means that we choose two members of the ith generation at random as parents, and then independently choose to use the first or second parent to donate the second and third partition value.

The mutation step is performed by simply adding a relatively small random value to the middle parameters of m_{ij1}, x_{ij11} and x_{ij21} as described here. The maximum size of the random value is determined by the number of points in [0,1] used to generate the 0th generation. In our case we used four points dividing [0,1] into thirds. We choose random values r_{ij11} $r_{ij21} \sim$ unif((-1/6,1/6)), since starting points are 1/3 apart. In addition to these deviation values, we calculate two more random values s_{ij11} and s_{ij12}, iid, with Prob(s=0) = .6 and Prob(s=1) = .4 (changing these probabilities will alter the chance of mutation). We then create $m_{ij2} = (0,x_{ij12},x_{ij22},1)$, where $x_{ij12} = x_{ij11} + r_{ij11}*s_{ij11}$ and $x_{ij22} = x_{ij21} + r_{ij21}*s_{ij21}$.

Let d_i, $1 \le i \le n$ be the frame data that we will use as our regressor, sorted in ascending order. Let k1 be a positive integer chosen such that $d_{k1} < x_{ij12} \le d_{k1+1}$ and similarly k2 a positive integer such that $d_{k2} < x_{ij22} < d_{k2+1}$. Let $x_{ij13} = d_{k1+1}$ and $x_{ij23} = d_{k2+1}$, then $m_{ij3} = (0,x_{ij13},x_{ij23},1)$.

We now wish to select the n_g members of the $(j+1)^{th}$ population. We have $2*n_g$ potential members, $m_{ij1} \le i \le n_g$ and $m_{ij31} \le i \le n_g$. For each of these $2*n_g$ potential members we calculate the MSE of the regression fit obtained from the sample data and the partition of [0,1] determined by the potential member. Call this MSE mse_{ij} for $m_{ij1} \le i \le n_g$ and mse_{ij3} for $1 \le i \le n_g$. We then select for $m_{i(j+1)}$ $1 \le i \le n_g$ the n_g potential members with the

smallest MSE's. The algorithm then returns to the recombination stage until the desired number of iterations has been met.

4.1 Stratification Based on the Regression Model

When we wish to stratify based on the regression model there are no modifications to the algorithm described above. The data set used plots the total pay for a unit in the 2002 Census with its value in the 2007 Census; if a unit was not present in both censuses it is excluded from the analysis in this paper; in practice we would simply handle these units in a separate stratum. We assume that the stratification based on this data set will be the same as the stratification that would be obtained if we were to use the full data from 2009.

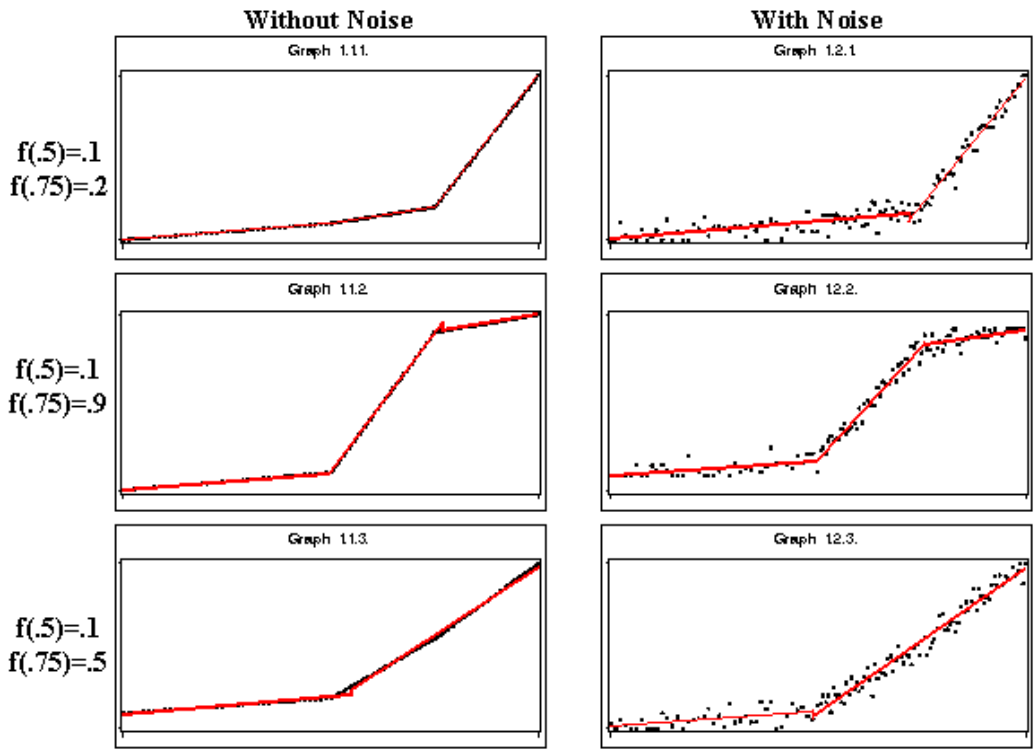
4.2 Stratification Based on Size

For size-based stratification we use a different data set as input into our algorithm. The data set is based on the 2007 Census and our size-based stratification is only performed on the PPS portions of the frame. We then sort the units within each initial stratum in ascending order of total pay. The x value for a unit becomes its cumulative percent of the number of units in the stratum, and the y value becomes its cumulative percent of the sum of total pay within the stratum.

5. Results

The plot below shows the model fitted to simulated data generated from a 3-piece linear model. Three different generating models were used, each one was defined by the value of f(x) at x=0.5 and x=0.75, with f(0)=0, f(1)=1, and linear interpolation between these points. The top row has f(.5)=.1, f(.75)=.2, the center row has f(.5)=.1, f(.75)=.9, and the bottom row has f(.5)=.1, f(.75)=.5. The left column uses data generated from the model with no noise, and the right column uses data generated from the model, plus iid terms coming from N(0,.05). The black dots (which form a line without the presence of noise) show the ‘true’ synthetic data, and the red lines are the model fit to the data.

Graph 1: Model Fit to 3-piece Synthetic Data, Without and With Noise



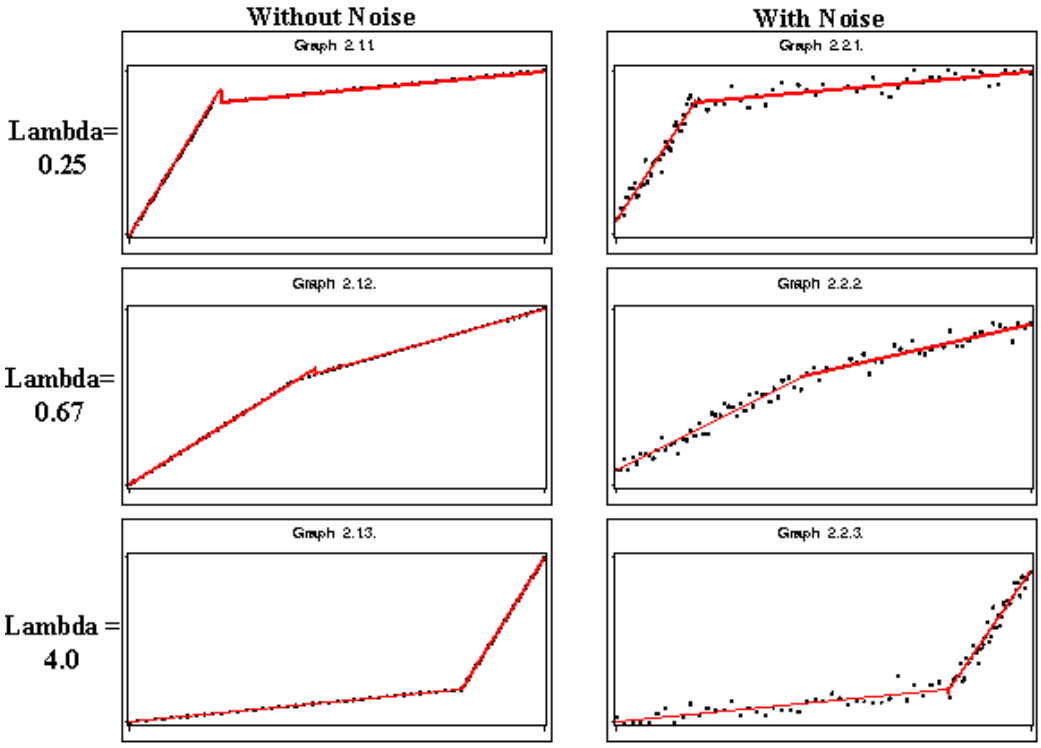
Source: Simulated data for illustrative purposes only

Note that although the two lines may be difficult to distinguish in the left, noiseless, column that both lines are plotted; they simply are plotted nearly on top of each other.

Visually, the fit for the second row is good both with and without noise. The upper left model shows a good fit as well. The remaining three models are fit with only two pieces. Considering the small visual change in the slopes, however, such a fit does not seem unreasonable.

The plot below shows the model fitted to simulated data generated from a 2-piece linear model. Three different generating models were used of the form $f(x) = (1/\lambda)*x$ for $0 \leq x \leq \lambda/(1+\lambda)$ and $f(x)= (\lambda)*x + 1 - (\lambda)$ for $\lambda/(1+\lambda) \leq x \leq 1$.The top, center, and bottom rows use a λ value of 1/4, 2/3, and 4 respectively. The left column uses data generated from the model with no noise, and the right uses data generated from the model, plus iid terms coming from N(0,.05). Again black dots represent the synthetic data and red lines the model fit. Once more the noiseless lines are plotted nearly on top of each other.

Graph 2: Model Fit to 2-piece Synthetic Data, Without and With Noise

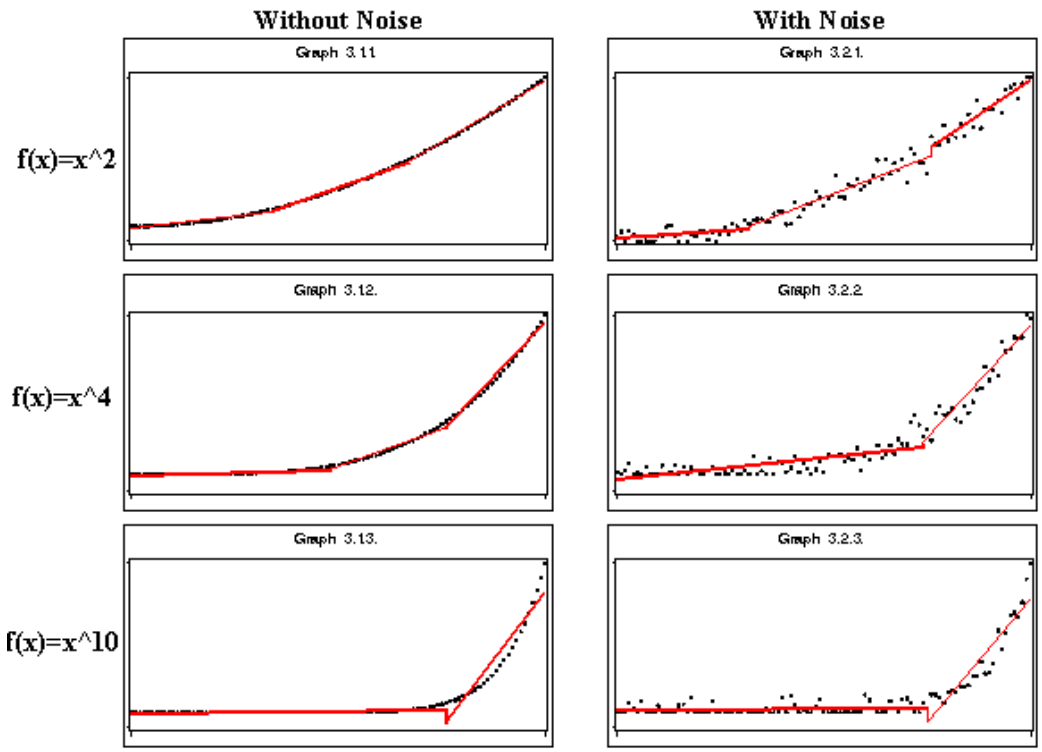


Source: Simulated data for illustrative purposes only

All of the models seem to be fit well, although the left column displays some small jumps that are less than desirable. We revisit this jump when we consider the next model.

The following plot shows the result of applying the algorithm to an exponential function defined on [0,1]. The first, second, and third rows are $f(x) = x^2$, x^4, and x^{10} respectively. The first column uses data generated from the model with no noise, and the second column uses data generated from the model, plus iid terms coming from N(0,.05).

Graph 3: Model Fit to Exponential Data, Without and With Noise



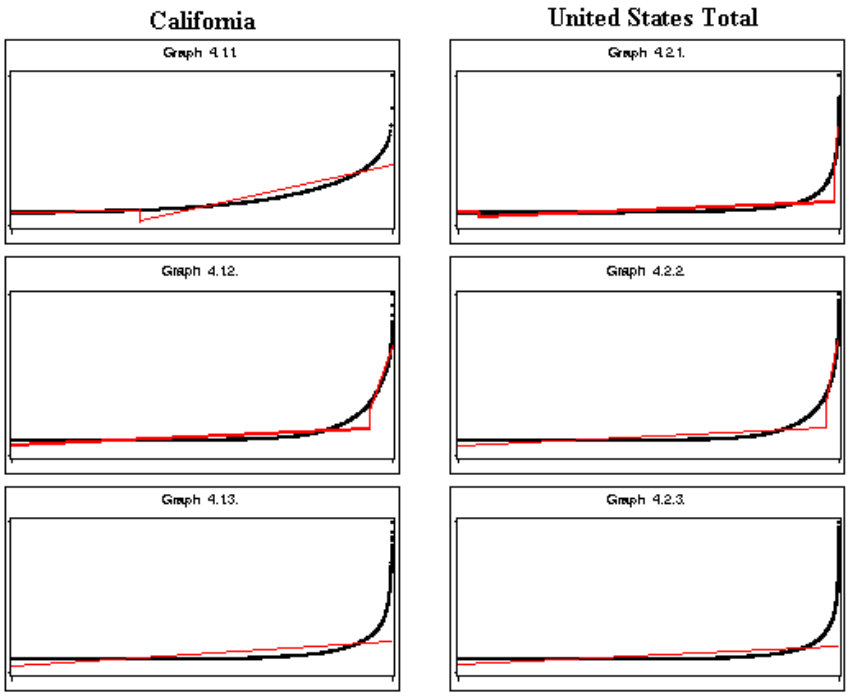
Source: Simulated data for illustrative purposes only

We see good fits for the first two rows; however row three displays an elbow where we would rather not have one.

Looking at some of the plots above, especially the bottom row, we see that introducing a constraint on the algorithm to force it to be continuous may improve our results, however due to time constraints this was not explored in this paper.

Below we see plots using real data, transformed to give us a way to stratify by size. The left column is California, the right is data for the entire United States of America. The top row displays data for cities and townships, the center displays all types of government combined, and the bottom row displays data for special districts. The horizontal axis of each plot shows the percent of units, sorted in ascending order by size (total pay), counted. The vertical axis shows the cumulative percent of total pay of the units counted so far. For example, from the bottom right graph we see that perhaps the smallest 90 percent of units account for only 10 percent of the total of total pay. As in all other graphs, the black lines and dots represent the (true) data that we wish to fit our model to, and the red lines are the model fit.

Graph 4: Cumulative Totals of True Data

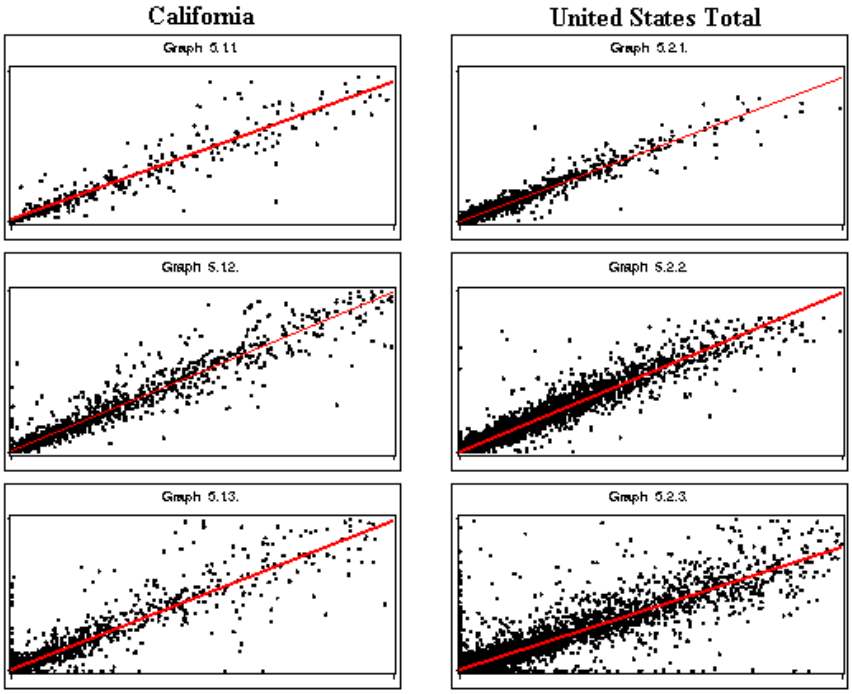


Row 1: Cities and Townships
Row 2: All Governments
Row 3: Special Districts

Source: U.S. Census Bureau, Annual Survey of Public Employment and Payroll

The data for the plots below come from the same cells as the previous set. However now we use slightly different data so that we can stratify based on our regression model. The units were still sorted in order of ascending value of total pay. The horizontal axis is the value of total pay for a unit in 2002, the vertical axis is the value of total pay for a unit in 2007; units which were present in only one of these years were excluded from consideration. The top row contains cities and townships, the center row contains all local governments combined, and the bottom row contains all special districts. The left column consists of data from California while the right column is an aggregate of all local governments in the nation. Once again black represents the true data and red the model fit.

Graph 5: Total Pay, 2007 Census vs. 2002 Census



Row 1: Cities and Townships
Row 2: All Governments
Row 3: Special Districts

Source: U.S. Census Bureau, Annual Survey of Public Employment and Payroll

Note that in California and at the national level, the algorithm failed to find multiple strata. Visually, total pay does not seem to divide into sized-based strata as far as the regression model is concerned so this result does not seem to suggest a flaw in the algorithm.

Table 1 shows the mean and coefficient of variance (CV) of the root mean squared error (RMSE) of the fit determined by the algorithm for the model specified by the table cell. These models correspond to Graph 2. For this model λ is a measure of the bendiness of the model, and σ a measure of the noise in the model. The first entry of each cell is the mean of the RMSE of 100 fits, and the second entry is its CV.

Table 1: Quality of Fit to 2-piece Synthetic Data, With Varying Noise Levels

G(x)=F(x) + N(0,σ)	F(x)=(1/ λ)*x when 0 ≤ x ≤ (λ /(1+ λ), F(x)= λ*x+1- λ when (λ /(1+ λ) ≤ x ≤ 1				
	λ = 0.25	λ = 0.67	λ = 1	λ = 1.5	λ = 4
σ = 0.01	0.01 (8.1%)	0.01 (7.4%)	0.01 (9.0%)	0.01 (8.0%)	0.01 (7.2%)
σ = 0.05	0.05 (8.8%)	0.05 (7.5%)	0.05 (7.7%)	0.05 (8.7%)	0.05 (8.7%)
σ = 0.10	0.09 (9.6%)	0.09 (8.2%)	0.09 (9.1%)	0.09 (7.5%)	0.09 (10.0%)

Source: Simulated data for illustrative purposes only

As shown in Table 1 above, the RMSE of the model fits tend to be very close to the standard deviation of the noise added. This suggests that we're getting as good of a fit as possible. The CV of our RMSE is fairly small (except in the case of σ = 0.00) so we feel that our algorithm exhibits good convergence behavior.

The values in the cells of Table 2 are the same as above, mean and CV and the RMSE, but we now consider the 3-piece linear model used for Graph 1.

Table 2: Quality of Fit to 3-piece Synthetic Data, With Varying Noise Levels

G(x)=F(x) + N(0, σ)	F(x): 3 piece linear function, with change points defined on F(0)=0, F(.5)=a, F(.75)=b, F(1)=1, where (a,b) is the column header.					
	(0.1,0.2)	(0.1,0.5)	(0.1,0.9)	(0.5,0.6)	(0.5,0.9)	(0.8,0.9)
σ = 0.01	0.01 (13.8%)	0.01 (20.8%)	0.01 (7.3%)	0.01 (8.5%)	0.01 (6.5%)	0.01 (6.6%)
σ = 0.05	0.05 (7.7%)	0.05 (6.5%)	0.05 (18.9%)	0.05 (7.9%)	0.05 (8.4%)	0.05 (6.7%)
σ = 0.10	0.09 (7.0%)	0.09 (7.5%)	0.08 (12.9%)	0.10 (6.2%)	0.09 (8.0%)	0.09 (7.7%)

Source: Simulated data for illustrative purposes only

In Table 3 we consider some real data. Since our initial analysis led us to believe that this methodology does not fit well to stratification based on the regression model, we consider only size-based stratification. Here we compare the efficacy of using our algorithm against the CSRF method. Three states, each divided into separate type categories, are

considered as well as the result of applying the method to the entire nation. For each cell we calculate the mean and CV of the RMSE.

Table 3: Comparison of Fits, Genetic Algorithm vs CSRF

Method Used to Stratify	Genetic Algorithm	CSRF-2
CA – Cities and Townships	0.06 (00.0%)	0.46
CA – Special Districts	0.07 (18.7%)	0.49
VT – Cities and Townships	0.03 (18.7%)	0.44
VT – Special Districts	0.11 (00.0%)	0.45
WI – Cities and Townships	0.06 (25.8%)	0.48
WI – Special Districts	0.08 (2.5%)	0.47
National – Cities and Townships	0.04 (29.4%)	0.49
National – Special Districts	0.06 (3.2%)	0.49

Source: U.S. Census Bureau, Annual Survey of Public Employment and Payroll

Compared with a 2-piece CSRF, our algorithm performs quite well. This is due partly to the fact that our algorithm allows for a third linear piece to be created and partly due to the simple fact that our algorithm is explicitly designed to fit any data well.

6. Conclusions and Future Research

This initial study into using genetic algorithms to partition our sampling frame shows promise. The algorithm itself seems to find a reasonable fit to a piecewise linear function, or a piecewise linear approximation to a non-linear function. When compared with the CSRF method our algorithm appears to perform much better according to the metric we used.

The next stage of the research would be to see how the different stratification approaches impact estimation. We hope to have chosen a final estimation methodology before doing this, but we can continue our research using standard methodologies if our production methodology is not available. As sampling and estimation are related to one another, we probably will not be able to finalize one without the other.

We intend to make further improvements of the algorithm as well. First, we would like to add conditions to ensure the continuity of the model fit generated by the algorithm. We hope this will remove some of the ‘elbows’ we see in the data and provide a better fit. We would also like to improve our stopping conditions. Presently we simply run the

algorithm for a fixed number of iterations. While this provides reasonable results, in many cases we could stop sooner and reduce the time spent running the algorithm. Finally we would like to add in a step to test if algorithm is attempting to divide a straight line into two pieces; this has not often happened in practice, but at some point we would look into the issue more rigorously.

References

[1] Barth, Joseph; Cheng, Yang; Hogue, Carma. Reducing the Public Employment Survey Sample Size. *2009 JSM Proceedings*, Survey Research Methodology Section. Alexandria, VA: American Statistical Association.

[2] Cochran, William G. *Sampling Techniques*, John Wiley & Sons, 1977

[3] Langdon, W. B., Poli, R., *Foundations of Genetic Programming*, Springer-Verlag, 2002