Harmonizing PUMAs across Time:
An Aggregation Algorithm to Achieve Minimally Acceptable Consistency

Jonathan Schroeder
David Van Riper
Minnesota Population Center – University of Minnesota, Twin Cities

Extended abstract for a poster presented at the
2016 Annual Meeting of the Population Association of America

The U.S. Census limits the geographic units identified in Public Use Microdata Samples (PUMS) to areas with populations large enough to protect confidentiality. Specifically, since 1980, no identifiable area may have fewer than 100,000 residents. To subdivide the entire U.S. into areas that meet this requirement without grossly exceeding it, the Census (with input from state data centers and planning agencies) defines custom geographic units, termed "Public Use Microdata Areas" (PUMAs) or, prior to 1990, "county groups." The spatial precision of these units is critically important for demographic research that aims to investigate or account for contextual effects, population-environment relationships, regional economies, or many other phenomena that operate at sub-state spatial scales. Unfortunately, when the Census updates PUMA boundaries after each decennial census (in order to implement new standards, to create smaller PUMAs in areas of population growth or larger PUMAs in areas of decline, and/or to support the changing interests of stakeholders in each state), the updates in many areas cause significant mismatches between the PUMA definitions over time. These mismatches present a major challenge for spatio-temporal analysis of microdata, making it impossible to hold the geographic units of analysis constant over time without additional data manipulation.

The Integrated Public Use Microdata Series project (IPUMS-USA, https://usa.ipums.org) at the Minnesota Population Center has addressed this problem by supplying "ConsPUMA" (consistent PUMA) codes, which group together PUMAs and county groups into sets that cover a generally consistent spatial extent across time. The original ConsPUMA variable identifies compatible sets of 1980 county groups and 1990 and 2000 PUMAs. It is available for 1980, 1990, and 2000 decennial census samples and for American Community Survey (ACS) samples through 2011, which was the last year for which the ACS PUMS use 2000 PUMA codes. Starting in 2012, ACS PUMS use 2010 PUMA codes instead. To provide ConsPUMA codes for the most recent ACS samples, we investigated several options for updating or supplementing the IPUMS ConsPUMA codes to include the new 2010 PUMAs.

This paper introduces and demonstrates the merits of a new algorithm we developed to group PUMAs from different times into sets that achieve "minimally acceptable consistency" according to a pre-specified, adjustable mismatch tolerance. We developed this algorithm as a more rigorous, objective alternative to the visual approach IPUMS-USA used to construct the original ConsPUMA codes. In that case, researchers used geographic information systems to inspect PUMA and county group boundaries and then "hand selected" sets whose boundaries were closely (if not exactly) in alignment. We are confident that this visual approach produced acceptably consistent sets overall, but when using a purely visual approach, it is possible to ignore some mismatches because of their small areas even though they in fact involve significantly large populations. Likewise, it is also possible that some visually constructed sets are needlessly large, combining PUMAs that have large areas of overlap even in cases where the population involved was insignificantly small. In contrast, the new automated algorithm should be more consistent, reliable, *and* efficient than the visual approach.

The general goal of the algorithm is to aggregate zones (in our case, PUMAs) from different, overlapping zonal systems to produce the *smallest possible* sets (in our case, "ConsPUMAs") that have "acceptably consistent" spatial extents across all zonal systems. The specific metric we use to determine whether two sets of zones are "acceptably consistent" is the *sum of percent omission error and percent commission error* in terms of zone populations. To compute this sum, we first select one set of zones (e.g., a group of 2010 PUMAs) to be the *reference set*. We can then compute how much a second "assigned" set of zones (e.g., a group of 2000 PUMAs) deviates from the reference set by computing the percent omission error (the percent of the total population of the reference set of zones that does not reside within the assigned set) and the percent commission error (the percent of total population of the assigned set that does not reside within the reference set). We sum these two statistics to obtain the final mismatch score.

To simplify our process, and to focus our efforts on the most urgent need, we have initially generated new ConsPUMAs that bridge *only* 2000 and 2010 PUMAs, and our algorithm, accordingly, is designed to operate on exactly two zonal systems. The algorithm may, however, be easily extended to include three or more zonal systems by progressively harmonizing zonal systems two at a time. For example, after producing ConsPUMAs that harmonize 2000 and 2010 PUMAs, the same algorithm can be used to harmonize the 2000-2010 ConsPUMAs with 1990 PUMAs, etc.

To compute mismatch scores for the 2000-2010 ConsPUMAs, we first define each ConsPUMA to match a reference set of 2010 PUMAs, and we then estimate both the 2000 and 2010 populations of each ConsPUMA's intersections with 2000 PUMAs by summing the populations of all 2000 and 2010 census blocks that have their geographic centers in each intersection. We obtain geographic definitions of blocks and 2010 PUMAs from the National Historical Geographic Information System (NHGIS),[1] which derives the boundaries from the Census Bureau's 2010 TIGER/Line files.[2] We construct 2000 PUMA boundaries by aggregating NHGIS's 2000 block polygons, using the Missouri Census Data Center's MABLE/Geocorr 2000 geographic correspondence engine[3] to determine which 2000 blocks comprise each 2000 PUMA. (For Puerto Rico, which is not covered by MABLE/Geocorr, we obtain 2000 PUMA boundaries from the Census Bureau's 2009 TIGER/Line files[4] and adjust a few boundaries to align with corresponding features in the 2010 TIGER/Line files.) We obtain block populations from 2000 and 2010 census summary files via NHGIS.

Given two input zonal systems—in our case, 2000 and 2010 PUMAs—the general concept of the algorithm is to begin with *no* aggregations—i.e., treat each 2010 PUMA as its own ConsPUMA—and then iteratively merge zones into sets, and in turn merge sets into larger sets, until each set's sum of mismatch errors falls below the specified tolerance. In our setting, the algorithm proceeds as follows:

1. Initialize the set of all ConsPUMAs to correspond exactly to the set of all 2010 PUMAs
2. Aggregate any set of ConsPUMAs where each ConsPUMA has a majority of its population in the same 2000 PUMA
    - This step is not essential, but acts as a time-saving "first assumption" that rapidly aggregates the most suitable aggregation candidates
3. Assign each 2000 PUMA to the ConsPUMA in which a majority of the 2000 PUMA's population resides

[1] https://nhgis.org
[2] https://www.census.gov/geo/maps-data/data/tiger-line.html
[3] http://mcdc2.missouri.edu/websas/geocorr2k.html
[4] https://www.census.gov/geo/maps-data/data/tiger-line.html

    - If a 2000 PUMA does not have a majority in one ConsPUMA, leave it unassigned
4. Compute the mismatch (sum of omission and commission errors) between each ConsPUMA and the 2000 PUMAs assigned to it
    - Compute the errors according to both 2000 and 2010 populations separately *and* again for the average of the 2000 and 2010 populations
5. Identify aggregation candidates among the currently defined ConsPUMAs
    - An aggregation candidate is a pair of ConsPUMAs where both share population with the same 2000 PUMA and where at least one of the ConsPUMA's degree of mismatch remains above the specified tolerance in terms of *either* 2000 or 2010 population
6. For each potential aggregation…
    - Assign 2000 PUMAs to the merged set (as in step 3)
    - Compute the potential mismatch (as in step 4)
    - Measure the potential mismatch decline as the average of the 2 ConsPUMAs' prior mismatch scores minus the new mismatch score for the merged pair
    - In this case, we use only the mismatch scores for the *average* of the 2000 and 2010 populations
7. Merge the pair of ConsPUMAs for which the mismatch score would decline most
    - Because 2000 and 2010 PUMAs have no significant areas of overlap across state lines, we shorten the process by identifying the largest potential decline *in each state* and merging all states' best candidate pairs at once
8. Repeat steps 3 through 7 until all ConsPUMAs have acceptable mismatch scores

We have implemented this procedure in MySQL and use it to produce ConsPUMAs with a mismatch tolerance of 1%. The output ConsPUMA codes are now available for IPUMS-USA samples from 2000 through 2014, and we will add them to forthcoming ACS samples as well. Our poster provides summaries and illustrations of how these ConsPUMAs differ in spatial precision and degree of mismatch relative to a solution based solely on visual inspection and another non-iterative solution that simply merges any pair of PUMAs that have a "large" overlap with the same PUMA from another year. We find that both the visual approach and the non-iterative solution produce several instances of unnecessarily large ConsPUMA sets, and that the visual approach also fails to aggregate some cases where the areas of PUMA intersections are small but the population involved is not.