Confidentiality risks of releasing measures of data quality

Jerry Reiter
Department of Statistical Science
Duke University
jerry@stat.duke.edu

General setting

- Original microdata, O
- Candidate release, M (1:1 link with O).

- Provide some measure of quality of M for specific analyses, labeled FM.
- Released dynamically for analyses submitted to a “verification server.”

Possible measures (think regression coefficients)

- *Overlap in confidence intervals*
  Compute 95% CI for Q with O;
  compute 95% CI for Q with M;
  measure overlap in intervals.

- *Distance between Q(M) and Q(O).*
  Absolute (relative) error.
  Percentage change in variance.

General confidentiality risks

- *Backsolving risk*
  If FM is exact, analyst might determine elements of O from FM and Q(M).

- *Prediction risk*
  If utility measure indicates small difference, analyst might closely estimate elements of O from FM and Q(M).

Example of attacks: Setting

- Suppose agency uses 3% data swap of five categorical quasi-identifiers, X.
- When record selected for swapping, all of its X is swapped with the X for another record.
- Other values, Y, are not swapped.

Example continued

- User asks for quality of

  - coefficients in linear regression of Y on X.
  - sample mean of Y.
  - mean of Y for small subpopulation defined by X.
  - contingency table analysis with X.

- Fidelity measure is confidence interval overlap (not coarsened). Perfect overlap defined as 1. No overlap defined as 0.

Intruder attacks

- Find out which records were swapped:
  Try different records in M for some query involving a subset of the data. If FM = 1, assume records were not swapped.

- Find out values of swapped X:
  Find at least one not swapped record for each level of univariate X.   Add target record with swapped X. Let Q be frequencies of each cell for the selected records. Correct value is the one not equal to swapped value for which FM < 1.

Another example

- To previous scenario, employ top-coding to protect upper 1% tail of a continuous Y1. And, add noise to all values of some Y2.
- Users desire regressions, means, confidence intervals for quantiles.

Intruder attacks

- Order the records by values of Y1:
  Form a data set with one top-coded record and many not top-coded records. Let Q be sample mean and obtain FM. Repeat for all top-coded records. Order records by FM.

- Estimate values of Y1 in upper 1%:
  Repeat above strategy. Try values of true Y1 that result in recreation of FM.

Intruder attacks

- Estimate values of Y1 in upper 1%:
  Form group of not top-coded records for which Y1 is transformed so that it equals zero (or any one number). Add one top-coded record. Get FM for mean of Y1 for these values. Try values of true Y1 that result in recreation of FM.

- Same strategy applies for learning values of Y2 with added noise.

Reducing risks of these attacks

- *Limit what is released*

  Report something other than exact FM. Coarsen or add noise to measures before release.

  Do not release FM for some Q.

- *Limit what is answerable*

  Do not allow any copying or transposing data.
  Do not allow arbitrary transformations of data.

Limiting queries

- Minimum sample sizes for queries.

- Automatic feedback for set of common transformations, and make all others go through disclosure review.

  (Counter attacks based on unusual transformations that are unlikely to be legit analysis).

Coarsening FM measures

- Report FM rounded, for example to nearest .05 for CI overlap. Hard to know what values are safe and useful for generic Q.
- Add noise to FM measure. Same difficulty!
- One approach is to create “acceptable” bounds for each true value, and choose rounding/noise to ensure those bounds are feasible under an attack strategy.

Randomness in FM measures

- FM based on Q(M) and Q(O-), where O- built by

  randomly delete k records from data used in Q(O).

  resample deleted records with replacement (random seed defined by Q).

- FM not true except by random chance.
- Adds large noise to Q with small sample sizes and small noise to Q with large sample sizes.

Where to go from here

- Evaluate resampling strategy
- Evaluate how much noise infusion/aggregation to the FM to defeat the attacks against topcoding or noise.
- Begin to develop system.