Data Swapping: A Risk–Utility Framework and Web Service Implementation

Shanti Gomatam, Alan F. Karr, Chunhua "Charlie" Liu and Ashish P. Sanil[1]
National Institute of Statistical Sciences, Research Triangle Park, NC 27709-4006, USA
{sgomatam, karr, cliu, ashish}@niss.org
www.niss.org/dg

Abstract

We describe a risk-utility framework for selecting among swapped data releases, NISS WebSwap–a Web service that performs data swapping, and the NISS WebSwap graphical user interface.

1 Introduction

Data swapping [2] is a statistical disclosure limitation (SDL) technique that works at the microdata (individual data record) level. Confidentiality protection is achieved by selectively modifying a fraction of the records in the database by switching a subset of attributes between selected pairs of records. In this paper, we present: (1) A risk–utility (RU) framework for selecting among candidate swapped releases corresponding, for example, to different choices of the swapping attributes and swap rate; (2) NISS WebSwap, a Web service that performs data swapping; and (3) The NISS WebSwap graphical user interface (GUI), which will be embedded in a data swapping toolkit (DSTK) currently under development at the National Institute of Statistical Sciences (NISS).

2 The RU Framework

Data swapping makes it impossible for an intruder to be sure of having identified an individual or entity in the database, because no record is certain to be unaltered. At the same time, the data are distorted, decreasing their utility for purposes such as statistical inference. Implementation of data swapping entails selection of swap attributes, the swap rate (fraction of records for which swapping occurs) and, possibly, constraints on unswapped attributes (Gomatam & Karr, 2003; Gomatam, *et al.*, 2003).

In our risk-utility framework, each candidate release is characterized by numerical values of disclosure risk and utility. See §5 for examples. A statistical agency would like to select a release that has both minimum risk and maximum utility, but ordinarily higher utility entails higher risk. Nevertheless, not all releases are sensible: any release dominates all other releases that have both lower utility and higher risk, so that the choice should be made from the *frontier* (in economics, the *efficient frontier*) of undominated releases. Figure 1 illustrates the frontier in a setting where utility is the absence of distortion. Selection of a release on the frontier can be done by assessing the risk-utility balance subjectively or quantitatively, by means of an objective function that relates risk and utility.

[1]Support for the research was provided by National Science Foundation grants EIA–9876619 and EIA–0131884 to NISS and by the National Center for Education Statistics (NCES).

3 The NISS WebSwap Web Service

NISS WebSwap is implemented as a free Web service (Cerami, 2002), and its Web Services Description Language (WSDL) description is available (NISS, 2003; Sanil, *et al.*, 2003), enabling users to access the swapping service using XML-based requests. Since NISS WebSwap is largely a research tool to study data swapping, we provide a GUI-based client (see §4) to access the service and perform the swapping.

The entire NISS WebSwap application consists of the client, which communicates with a Java *Servlet* running on the NISS Web Services server, which in turn passes extensible markup language (XML) requests to and from the Web Service, also running on the NISS server. Technically, the Servlet is the client of the Web service; for general users, the GUI client should be as lightweight and portable as possible, which is accomplished by inserting the Servlet layer to handle the XML messaging.

The client is a Java application that constructs and transmits the specifications and data to the Servlet using Java's remote method invocation (RMI) protocol. The Servlet is a (server-side) Java application that runs within a servlet container on the NISS server. The Servlet, using libraries that accompany Sun's Web services toolkit, bundles the data and specifications into an XML request that it transmits, using the simple object access protocol (SOAP), to the Web service. The Web service receives the request from the Servlet, extracts the data, carries out the requested swapping, and rewraps the swapped data and auxiliary information as an XML response that is passed back to the Servlet. Finally, the Servlet unbundles the XML response and transmits the swapped data back to the client on the user's machine.

Details of the swapping algorithm employed by NISS WebSwap appear in Sanil, *et al.* (2003). The algorithm involves random selection of pairs of data records for swapping, so different runs, even with the same parameters, produce different release candidates.

Because of security considerations, the current version of NISS WebSwap cannot be run on confidential microdata. Instead, a statistical agency would implement NISS WebSwap as a Web service on its Intranet. Implementation as a Web service is a particularly efficient mechanism for providing the core swapping functionality for analysts within a statistical agency who are engaged in SDL tasks. The general data swapping policies can be conveniently maintained in one central place, and the users can easily access the service from multiple computing platforms, as well as from within their own applications.

4 The NISS WebSwap Client

Here we describe the NISS WebSwap client software. A complete description is contained in Sanil, *et al.* (2003). The client distribution, consisting of a Java JAR (Java Archive) executable, demonstration files and user documentation, may be downloaded from the NISS Web site (NISS, 2003). A Java 2 runtime environment compliant with the Java 2 Platform, Standard Edition (J2SE) version 1.3.1 or 1.4.x must be installed on the user's computer.

Input Files. Two input files are required by NISS WebSwap: a comma-separated value (CSV) *data file* containing the microdata records (which must have unique identifiers) and a *description file* containing corresponding metadata—attribute names and types.

The Specifications File. The specifications file is an ASCII text file produced automatically by the NISS WebSwap client, and contains (1) The number of records in the data file; (2) Names of the data file, description file, log file, output file and specifications file; (3) The swap rate; and (4) The *swapping*

specification—for each attribute, whether it is to be Swapped, must remain Fixed, must Differ between any pair of records that are swapped, or is not constrained (Other).

Output Files. NISS WebSwap produces and transmits to the user's computer a CSV *output file* containing the swapped microdata and a *log file* containing details of the swapping process, such as the number of swaps performed.

NISS WebSwap User Interface. The NISS WebSwap GUI is used to create or edit the specifications file required by the NISS WebSwap Web service, by allowing the user to specify file names and construct the swapping specification. The basic unit for NISS WebSwap is the *project*, which is defined by its specifications file and associated data and description files. Existing projects are loaded via the `Open` item on the `Project` menu, and new ones are created with the `New` item.

An `Open` command loads an existing specifications file (*.specs), and provided that the data and description files that it requires exist and are compatible, allows the user to edit the output and log file names and swapping specification. The `New` command creates a project by selecting the underlying data file (*.orig), loads and verifies the associated description file (*.desc), and provides a window that allows the user to enter output file names and the swapping specification.

Once the swapping specification is complete and written to the specifications file, the user invokes the NISS WebSwap Web service by clicking on a Swap button in the main window of the GUI. The main window shows various information messages, such as those resulting from a successful swap, including the names of the output files. There is no direct capability to view these files, but they may be examined with any file viewer. Upon exit from NISS WebSwap, the user has the option to save the cumulative contents of the main window to a session file, or to append the contents to an existing session file.

5 Risk and Distortion: Calculation and Display

For categorial data, our initial measure of disclosure risk is the proportion of *unswapped* records in small count cells in the table created from post-swap data:

$$\text{Risk} = \frac{\sum_{C_1,C_2} \text{Number of unswapped records}}{\text{Total number of unswapped records}},$$

where C_1 and C_2 are the cells in the full data table with counts of 1 and 2. This measure is derived from the n-rule (in our case, $n = 3$) in the SDL literature: database uniques and near uniques, which appear in small count cells, are at greater risk of re-identification.

We measure (dis-)utility by comparing the pre- and post-swap databases: the "closer" these are, the higher the utility. For clarity, we term lack of closeness *distortion*. Specifically, building on Gomatam & Karr (2003), we use Hellinger distance between the pre- and post-swap data tables to quantify distortion. Measures of utility that are tied more closely to uses of the data for statistical inference are a topic for future research.

We have also constructed an interactive visualization tool for risk, distortion and frontiers arising from multiple choices of swapping attributes and swap rate, which is illustrated in Figure 1 for data from the Current Population Survey (CPS). The associated study, described in detail in Gomatam, *et al.* (2003), considered three swap rates and all choices of one and two swapping attributes in an 8-attribute, 48,842-element categorical database. There is total of 108 (3 rates $\times$ 36 [= 8 one-attribute + 28 two-attribute]

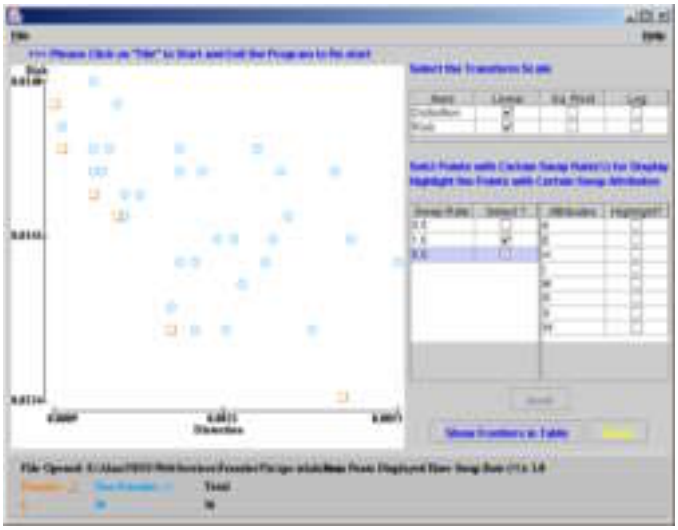

Figure 1: Visualization tool for risk, distortion and RD frontiers. The cases shown are 8 one-attribute and 28 two-attribute 1% swaps from an 8-attribute categorical database. Candidate releases on the frontier are indicated by squares.

choices of swap attributes) candidate releases. Such a study is conducted by running NISS WebSwap 108 times, and then calculating risk and distortion for each, using standalone programs written in C. The risk and distortion calculations require both the input file and the output file. Finally, the results are fed to the visualization tool, a Java application that provides numerous selection and drill down capabilities.

The DSTK mentioned in §1 will incorporate all of these components in one package, in modular form, together with scripts to link them.

References

[1] E. Cerami. *Web Services Essentials.* O'Reilly & Associates, Sebastopol, CA, 2002.

[2] S. Gomatam and A. F. Karr. Distortion measures for categorical data swapping. *J. Official Statist.*, 2003. Submitted for publication.

[3] S. Gomatam, A. F. Karr, and A. Sanil. A risk-utility framework for categorical data swapping. *J. Official Statist.*, 2003. Submitted for publication.

[4] National Institute of Statistical Sciences. NISS WebSwap, 2003. Available on-line at www.niss.org/WebServices/dg/WebSwap.html.

[5] A. P. Sanil, S. Gomatam, A. F. Karr, and C. Liu. NISSWebSwap: A Web Service for data swapping. *J. Statist. Software*, 8(7), 2003.