Use of Synthetic Data in Testing Administrative Records Systems

K. Bradley Paxton and Thomas Hager

ADI, LLC
200 Canal View Boulevard, Rochester, NY 14623
brad.paxton@adillc.net, tom.hager@adillc.net

Executive Summary

Synthetic data from ADI in the form of Digital Test Decks® have been successfully used in the U.S. Census for over a decade to test and evaluate data capture systems. As we began to leverage this technology into other areas, we observed that the nature of the data by itself had intrinsic value, quite apart from the particular method used to deliver it: paper, image or electronic data. We have found that large, robust, realistic, longitudinally interconnected and statistically valid data sets are extremely effective for development testing of a wide range of classification systems, from health records to intelligence gathering. Over the last few years, we have developed a novel approach generating of this data called the Dynamic Data Generator™[1]. This generator can create very large and complex data sets designed for testing which we call a Great Automated Model Universe for Test (GAMUT), from which suitably formatted data streams may be extracted to suit particular test objectives.

In particular, administrative records systems, which are becoming ubiquitous in government agencies as well as large corporations, are a natural candidate for this new and unique testing technology. A suitable GAMUT, coupled with experience in classification systems testing and evaluation, makes it possible to test and evaluate administrative records systems more efficiently than ever before. In this short paper, we focus on testing the record linkage aspects of administrative records systems.

Current testing methods for record linkage systems are inadequate for rapid, high-quality development work. The exclusive use of "real" data for testing creates security problems, and does not allow for solid quantitative results since the Truth of the data is not known. De-identified data is almost useless for good testing, as critical features of the data that are needed by the classifier to perform properly are absent. Manually created small data sets are costly and insufficient to test today's complex classification systems, and cannot provide scalability proof of concept.

The use of carefully engineered synthetic data sets *designed for test* and for which the *Truth is known* enables more cost-effective, precise and efficient testing to be done rapidly, improving system quality and reducing program risk and cost in developing record linkage systems.

In this paper, we illustrate a sample data set that was designed to test a record linkage system, how it could be used to test, and finally, show how one can perform timely, cost-effective and precise scoring of system matching results, including analysis of true and false positives to select optimal classifiers or choose optimal classifier settings.

[This paper was published in the Federal Committee on Statistical Methodology (FCSM) 2012 Conference Proceedings]

Testing Record Linkage with a GAMUT of Engineered Synthetic Data
Using our Dynamic Data Generator™ (DDG), it is possible to create a useful model universe we call GAMUT (Great Automated Model Universe for Test), as diagrammed in Fig. 1.

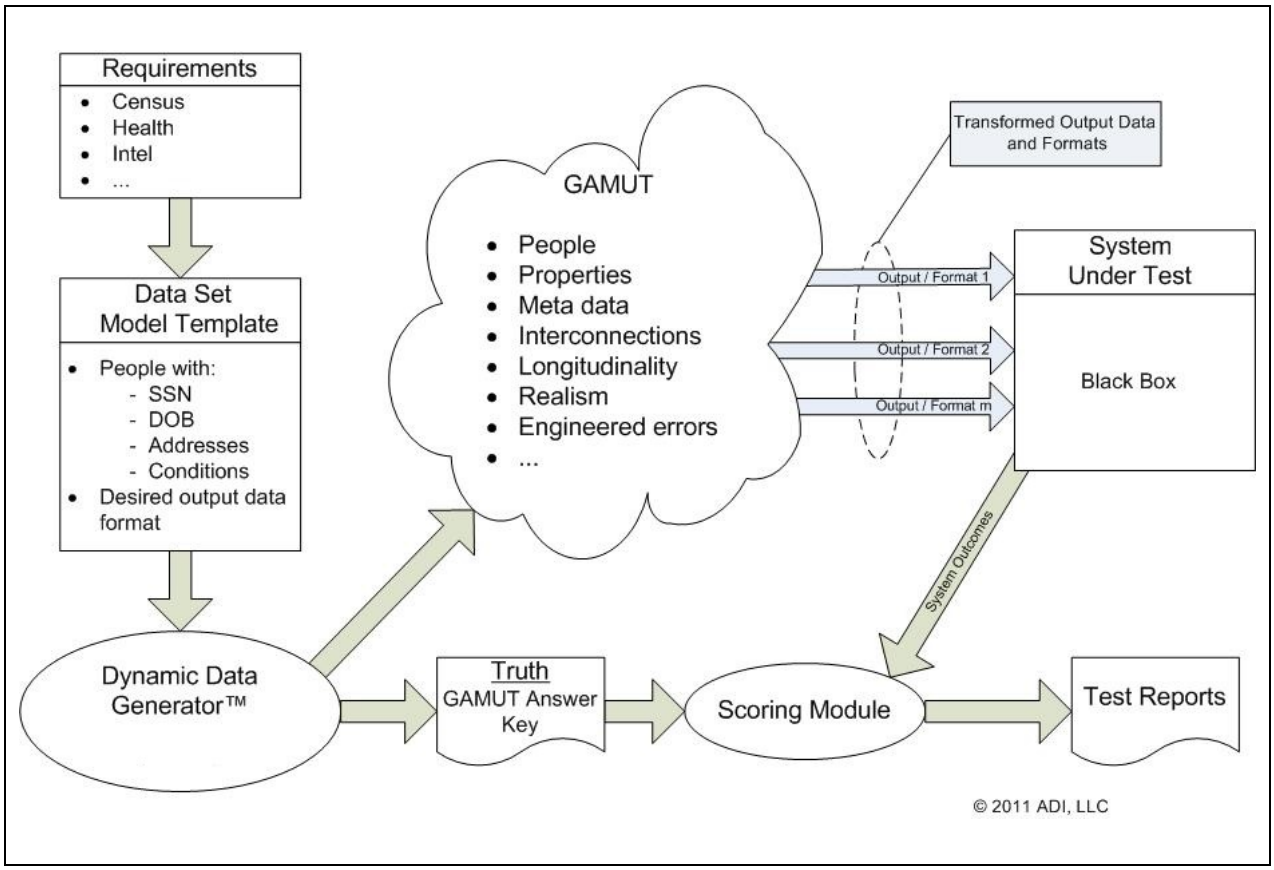

Fig. 1 – GAMUT: Great Automated Model Universe for Test

We begin by considering the requirements of the System Under Test (SUT). In this paper, we will describe a hypothetical census-type Record Linkage system for using administrative records. The primary entities to be created are determined; often, the entities are persons or households. The test plan objectives guide the design of the data set model template needed to run the DDG™, along with the necessary data fields to be populated. Additional considerations may include the properties, metadata, the desired type of interconnections the data should have, and the quantity and type of engineered errors. Then we run the DDG™ and create the appropriate GAMUT, extracting the desired test data sets in the formats required by the SUT and producing the Truth files necessary for test result evaluations.

After the test is run, the SUT outputs are suitably compared to the Truth to score the test and produce quantitative metrics that allow for tuning, testing, evaluation and comparison of classifiers and threshold parameters.

An Example for Census Administrative Records Testing
We now get more specific about our example, in this case involving a hypothetical census records linkage system that ingests two data streams (see Fig. 2).

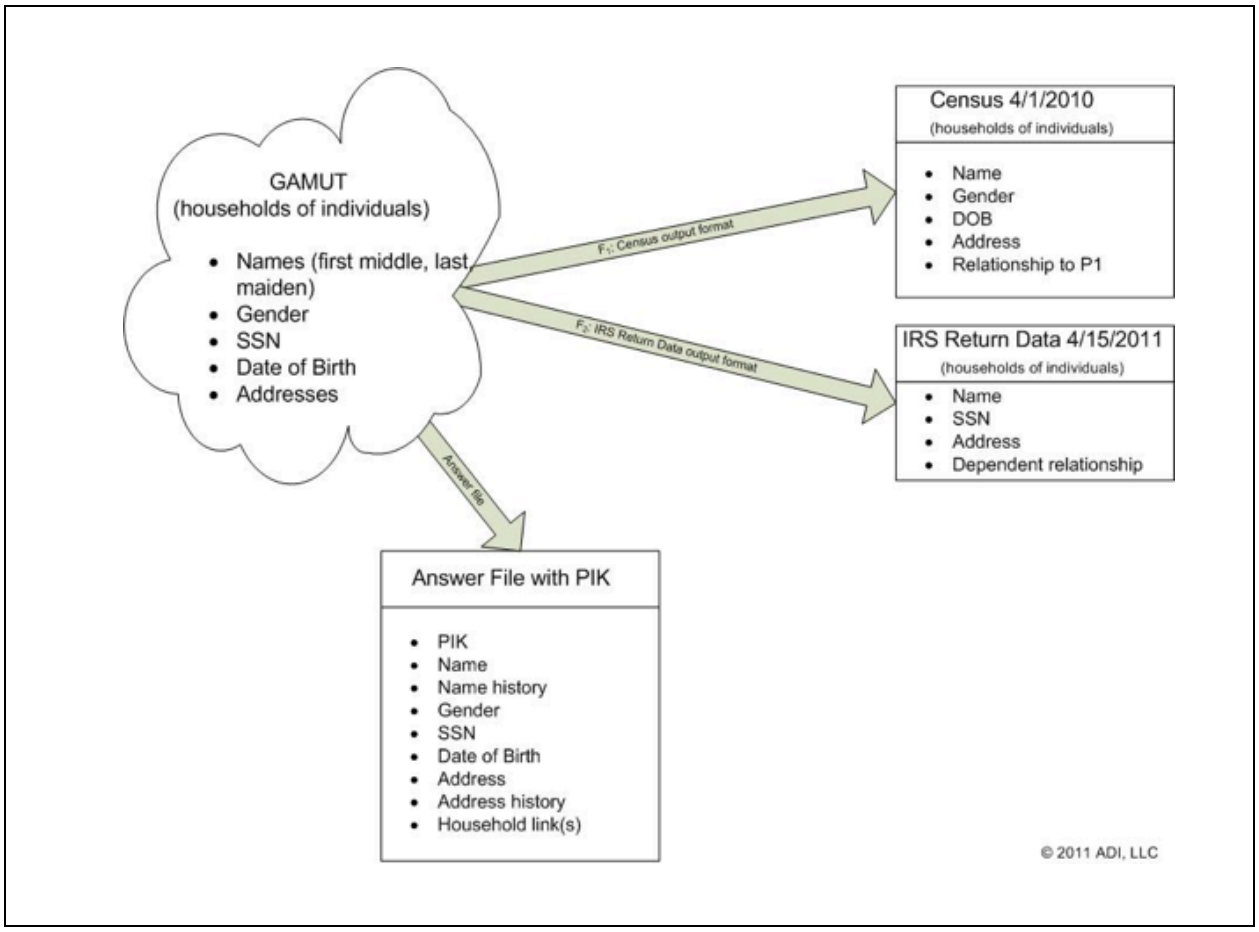

Fig. 2 – Data for an Example Census Administrative Records System Under Test

This example's administrative records system has as its purpose to identify people in one stream of data with people in a second stream of data. We call the first data stream F_1; we call the second data stream F_2. This example classification system's primary function is to attempt to find matches between F_1 and F_2, for each person in F_1.

Our first data stream is a census data stream (F_1), containing household data characteristic of data collected on census forms, such as head of household, name, gender, address, phone number, age, date of birth, race, ethnicity, date and relationship to other household members. The second data stream we define as a tax form data stream (F_2), which contains name, Social Security Number (SSN), address, relationship of dependents or spouse, income, date, etc.

You will note that these two data sets overlap somewhat in terms of names and addresses, but not completely, e.g., gender, date of birth, or social security numbers. This is but a part of what makes record linkage challenging.

There are many other reasons why such a matching function is difficult. One is that people's names are often written or typed in different ways that make it difficult for a computer to establish a match. For example, the full name KENNETH BRADLEY PAXTON may be (actually) found in different government and business documents as:

KENNETH BRADLEY PAXTON
K. BRADLEY PAXTON
KENNETH B. PAXTON
KEN B. PAXTON
K. BRAD PAXTON
KENNETH PAXTON
BRADLEY PAXTON
KEN PAXTON
BRAD PAXTON
K. B. PAXTON
K. PAXTON
B. PAXTON
…

These are but some of the basic ways a typical U.S. name might appear in real documents, but there may be a lot more variations depending on spelling errors, phonetic errors, OCR errors, keying errors, etc. They all could refer to the same person, or not; and possibly matching inferences could depend on other data, for example, addresses or dates of birth. Our realistic test data contain as many of these variations as are appropriate for testing, depending on the test plan.

Another reason matching is difficult is that people and households move, or change their residences at frequent intervals, something like 10% a year in the U.S. Another is that records may be duplicates or missing. Our realistic test data has duplicates, missing records, and connected longitudinal data, that is, realistic data that changes over time.

Another reason matching is difficult is the mere size of the data streams that are of interest. Two data streams of only a thousand records each, for example, produce a million possible "comparisons" for determining if there is a match. Our data is created in large numbers of records automatically, and so is useful for realistic load testing as is required to test scalability.

Test data suitable for testing administrative records matching systems must take these possible practical difficulties, and many more, into account. In addition, the Truth of the test data is produced from the GAMUT in such a way as to be useful in scoring outputs of the System Under Test (SUT). This solves a major problem with present-day testing of Record Linkage systems, namely that the matching Truth is not known.

These data streams are all produced from the original GAMUT, and so the Truth is exactly known as to which individuals match, even if it is very difficult (or even impossible) for the SUT to confidently predict a match. The truth files are associated with a Protected Identification Key (PIK) that is often used in record linkage systems to allow customers to easily link to other data. Also, for simplicity in this particular example, we will only attempt to match "head of household" people from F_1 with people in F_2, and we removed duplicates before linking.

In Fig. 3, we graphically display some example Truth for this binary classifier: the first person in F_1 matches the second person in F_2; the second person in F_1 does not match anyone in F_2; the third person in F_1 matches the fifth person in F_2; the fourth person in F_1 matches the first person in F_2, etc.

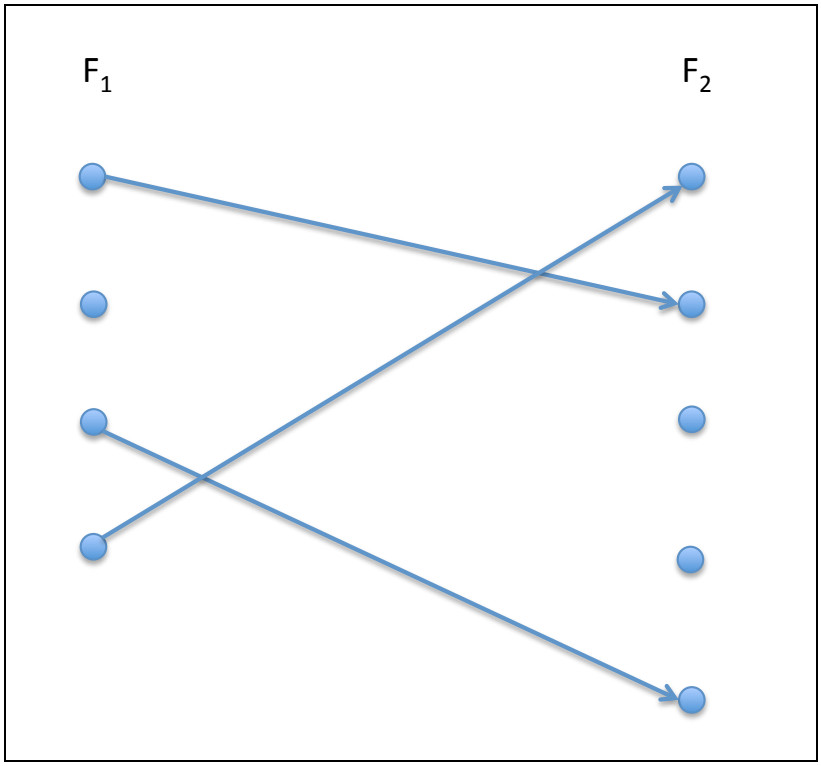

Fig. 3 – An Example of Test Matching Truth – Blue Arrows Indicate a Match

A major problem in testing with "real" data is that the Truth is *not* known, and it is very laborious and costly to determine it with manual methods. As a result, many organizations do not do a very thorough job of performance accuracy testing, and often only count the number of *predicted* matches found by the SUT. However, these predicted matches might either be correct or incorrect, as explained below.

When these test data streams are read into the SUT, and the predicted matches are produced, we can get four possible results for a binary classifier, graphically shown in Fig. 4: person one in F_1 is correctly matched to person two in F_2 (a *true positive* or TP); person two in F_1 is correctly not matched to anyone in F_2 (a *true negative* or TN); person three in F_1 is incorrectly not matched to anyone in F_2 (a *false negative* or FN); person four in F_1 is incorrectly matched to person three in F_2 (a *false positive* or FP). For each person in F_1, there is one of these four outcomes for a binary classifier, assuming duplicates are removed, as is the case in our examples below.

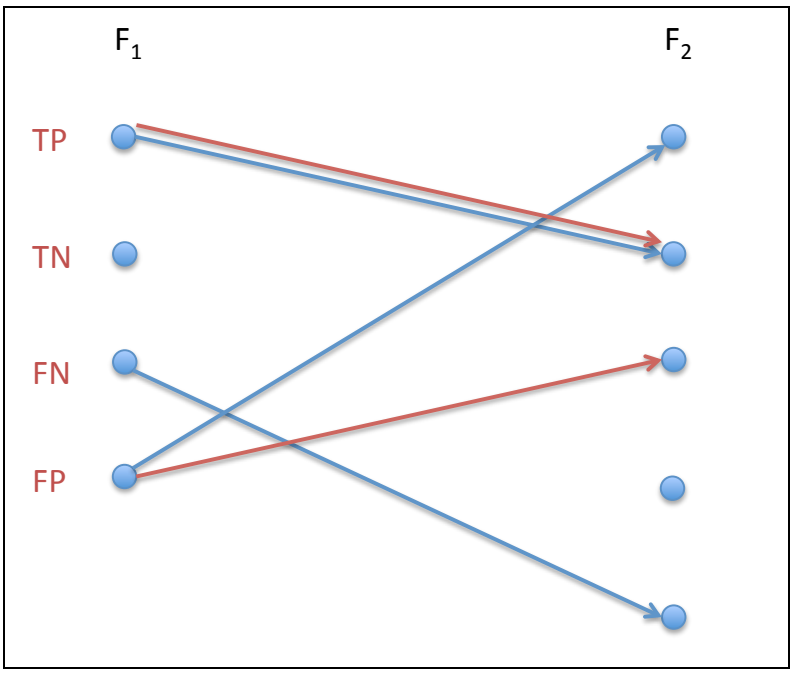

Fig. 4 – An Example of Test Matching Truth with Four Possible Binary Classification System Results

If you are testing with GAMUT data and related Truth as described herein, these quantitative outputs may be placed into a "confusion matrix" in order to more easily see how the SUT behaved, as shown in Fig. 5. Here, the Truth for positive or negative matches is defined on the rows; the predicted positive or negative matches is defined on the columns. The sum of the first row is defined as M, the total number of true positive matches (known from the Truth). In typical testing with real data, *this number is never known*. The total number of positive matches predicted by the SUT is defined as m (observed from SUT output). The total number of elements in the matrix is defined based on the number of elements in the two data sets and the nature of the test; in our example, we will make it the number of persons (head of household) in F_1. Finally, using the Truth, the precision c may be determined as the fraction of the predicted matches that are true positive matches, or c = TP/(TP + FP). When the matrix is completed, we have N = TP + FP + TN + FN.

		SUT Prediction	SUT Prediction	Row Sums
		Positive Match	Negative Match	
Data Truth	Positive Match	**TP** cm	**FN** M - cm	M
Data Truth	Negative Match	**FP** m(1 - c)	**TN** N – M - m(1 - c)	N - M
	Column Sums	m	N – m	N

Fig. 5 – A Confusion Matrix Suitable for Binary Classifier Evaluation

An Actual Test

We prepared a small synthetic data set for illustration of all this containing (only) 1,000 households in the GAMUT. There were 11 duplicates in the census data stream that were removed; and there were four missing census records, so the census data stream F_1 contained 985 unique households. For the confusion matrix, we find it convenient to be in an entity space rather than a comparison space, and so we choose N = 985. The tax file F_2 had 148 missing records (about 15% of people don't file tax), and so had 852 tax records. From the GAMUT Truth, we know there were 848 true positive matches (M = 848). There were a few engineered errors placed in the data set such as name morphing, some obfuscated fields, some households moves between data snapshots and so on, but this was not (intentionally) a difficult data set.

We have a record linkage research tool in our labs called Febrl[2], and we set it up to process our two small data streams under two sets of conditions: the first experiment (E_1) used five discrete comparisons - first name, last name, address, city and state; the second (E_2) used only first name, last name, city and state. (Note date of birth was not used as it is not on tax forms).

The first experiment (E_1) predicted 808 positive matches (m = 808), with three of them being false positives; and so, the precision was c = TP/(TP + FP) = 805/808 = 0.9963. The entire confusion matrix is shown below, along with some additional calculations, classically used in this type of analysis.

N	M	m	c
985	848	808	0.9963

	Prediction of S.U.T.		
	Pos	Neg	
Pos	805	43	848
Neg	3	134	137
	808	177	985

TPR	FPR	A	f
0.949	0.022	0.953	0.972

Fig. 6 – Confusion Matrix and Results for Experiment E_1

The additional calculations include the *True Positive Rate* (TPR = TP/(TP + FN), sometimes referred to as *recall*), the *False Positive Rate* (FPR = FP/(FP + TN)), the *Accuracy* (A = [M (TPR) + (N – M) (1 – FPR)]/N), and the *f-score* (f). The f-score is a harmonic mean of precision and recall, and is often handy when we get lots of True Negatives. Here, the accuracy was A = 0.953, which is reasonably good.

It is traditional to plot TPR vs. FPR and obtain a point on the Receiver Operating Characteristic (ROC) curve, shown below in Fig. 7.

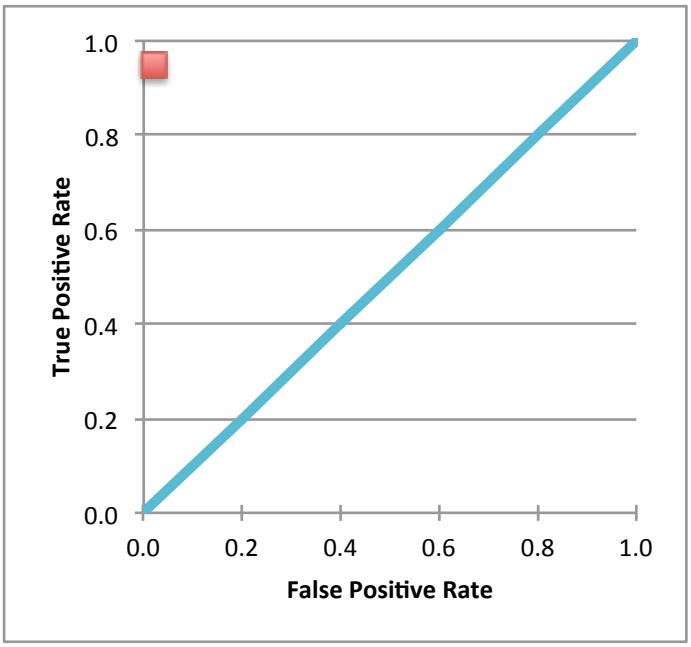

Fig. 7 – ROC Plot for Experiment E_1

Perfection on a ROC plot is at the upper left-hand corner at TPR = 1 and FPR = 0. This is another indicator that E_1 went pretty well.

When we ran the second experiment (E_2), leaving off the address comparison, we obtained the following, as shown in Fig. 8:

N	M	m	c
985	848	925	0.8843

	Prediction of S.U.T.		
	Pos	Neg	
Pos	818	30	848
Neg	107	30	137
	925	60	985

TPR	FPR	A	f
0.965	0.781	0.861	0.923

Fig. 8 – Confusion Matrix for Experiment E_2

Here, the accuracy is less than for E_1, indicating that E_2 was not as good.

When we plot the ROC data for E_2, we see it moved away from perfection (the upper left-hand corner) a significant amount.

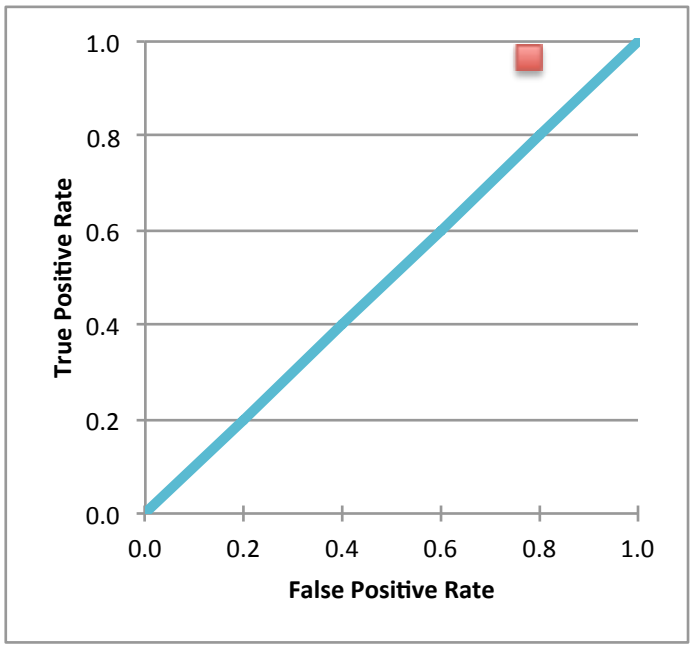

Fig. 9 – ROC Plot for Experiment E_2

Even though the second experiment (E_2) got more predicted matches (925 compared to 808), and got more True Positive matches (818 compared to 805), the overall results were worse than E_1. This is because of the significant increase in False Positives, which can only be calculated if the Truth is known.

This example demonstrates the potential power of using high quality synthetic data with known Truth for testing record linkage systems.

Contact one of the authors if you like to have us e-mail you the small data set used for this study, along with the truth files, and metadata describing the actual data features.

Conclusion

The use of engineered synthetic data sets *designed for test* and for which the Truth is known enables more cost-effective, precise and efficient testing to be done rapidly, improving system quality and reducing program risk in developing record linkage systems.

References

1 U.S. Published Patent (Pending), *System and Method for Rule-Driven Constraint-Based Generation of Domain-Specific Data Sets*, Filed 12 Mar 2010, Pub. No.: US 2011/0153575 A1

2 Peter Christen, *Febrl – An Open Source Data Cleaning, Deduplication and Record Linkage System with a Graphical User Interface*. In: ACM KDD, Las Vegas (2008)