Testing Record Linkage Production Data Quality

K. Bradley Paxton, ADI, LLC

(DRAFT v. 1.7)

Background

Record Linkage (RL) is used to find common entities (e.g., persons, households, or businesses) between pairs of data records in disparate data files. Once these links are found, an improved data set may be obtained by merging the matched entity data. This resulting improved data set could then be used for the appropriate business purpose or further examined by "data mining". If, however, the record linkage is done poorly, the "improved" data set might actually be worse than before.

Testing the production output data quality for record linkage systems is very difficult - most find it so difficult they barely do it at all. This means many practitioners of record linkage don't know precisely how well their system actually works, much less how to make it better. In this paper, we outline a way to use automation to enable the efficient measurement of record linkage data quality in production or in development testing using "real" data. We call our automated testing approach RLPDQ, which stands for Record Linkage Production Data Quality, and it is an extension of the PDQ system that was used successfully in the 2010 Census to measure data capture quality in forms processing (Ref. 1).

The Record Linkage Testing Problem

A typical record linkage system would input two files, F_1 and F_2, where the number of entity records in each file is N_1 and N_2 respectively. These files are input to the record linkage system (the system under test, or SUT), as shown in Fig. 1.

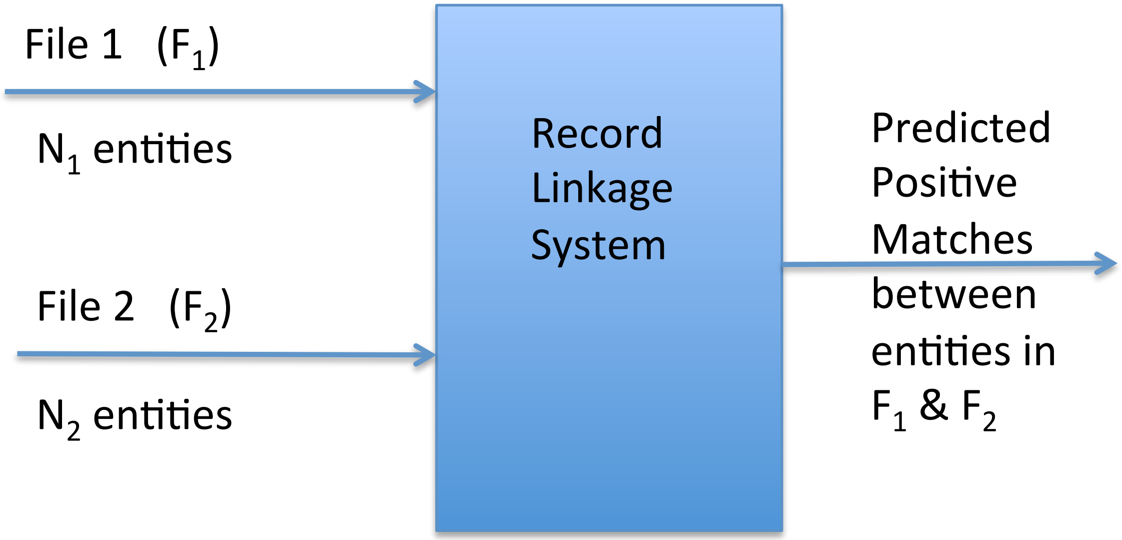


Fig. 1 – A Typical Record Linkage System

A typical record linkage test using “real” data is often done using two input files from actual sources; let’s say for our purposes here the two files represent Census-type data and Tax-type data. The record linkage system is run and a fraction of the total estimated possible positive matches is recorded, say, 90%. Then possibly an “improvement” to the record linkage system is made, and now 93% matches are obtained. This sounds at first like a better outcome, however, if the additional predicted positive matches are false positives, the outcome is actually worse. It is very difficult to measure false positives in record linkage tests with “real” data or in production because the “Truth” is not known.

In general, the predicted positive matches from any SUT will contain both true positives and false positives. The rest are predicted negative matches, containing both true negatives and false negatives. This gets a little confusing, and so in order to help explain it we employ a “confusion matrix”, as shown in Fig. 2. The rows of the confusion matrix are the data truth for both positive and negative matches and the columns are predictions from the SUT for both positive and negative matches.

		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. 2 – A “Confusion” Matrix for Record Linkage Testing Results

It is only when you can fill out numbers in all four boxes on the confusion matrix that you can actually say you understand how well your record linkage system is working. If you can't do that, not only are you unsure as to how well your system actually works, you are not sure what to do to make it better. In the confusion matrix the term usually called *precision* is donated by small c, the number of predicted positive matches is small m, the number of actual positive matches is large M and the number of elements all totaled in the confusion matrix is large N.

The things that you like (correct matches) are on the matrix main diagonal: true positives and true negatives. The things that you don't like (incorrect matches) are on the off diagonal: false negatives and false positives. Often, the false positives are called Type I errors, and the false negatives are called Type II errors. Which of these two types of errors you like the least depends on the nature of your record linkage objectives, and is usually related to the overall program “cost” of dealing with these errors.

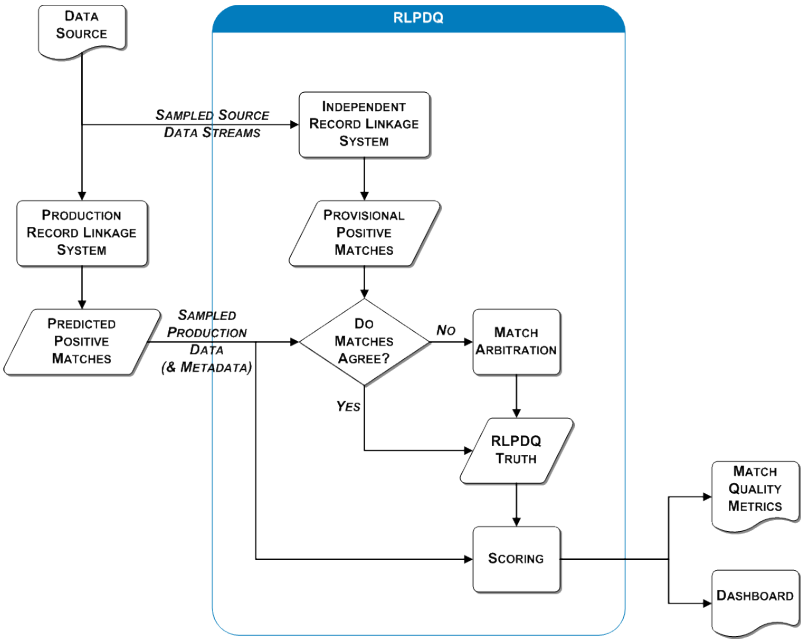
Basically the testing problem is this: testing record linkage systems with real data is extremely difficult, and it is expensive to obtain quantitative metrics like false positives and false negatives (Ref. 2). Further, if record linkage has errors, then serious consequences are possible, for example, medical records, voter registration records, and use of administrative records in future Census applications.

In classification system development and tuning, use of synthetic data allows testing to produce quantitative metrics as we previously showed in Ref. 3. The Production Data Quality (PDQ) system developed for census forms data capture (Ref. 1) is extensible to record linkage systems as we're indicating in this paper. In production, Record Linkage Production Data Quality (RLPDQ) can bring automation to bear on testing when doing record linkage with real data.

What Is RLPDQ?

In the RLPDQ block diagram in Fig. 3 we show a data source in the upper left-hand corner supplying data to the production record linkage (RL) system, and also sending the same data, possibly sampled, into the independent record linkage system. In order to achieve low measurement error, it is important to insure that the independent RL system is fundamentally different than the production RL system. This may be accomplished by choosing a different technology, with different software code, or different matching criteria, or preferably both. When this was done in the case of forms processing (Ref. 1), the PDQ measurement error was estimated to be less than 0.01%.

Both the production RL system and the independent RL system will then predict positive matches without knowledge of the other's inner workings. Matches not predicted to be positive by either system are therefore predicted to be negative by both engines; and these predicted negative matches are comprised of both true negatives and false negatives.



Data Source
RLPDQ
Sampled Source Data Streams
Independent Record Linkage System
Production Record Linkage System
Provisional Positive Matches
Predicted Positive Matches
Sampled Production Data (& Metadata)
Do Matches Agree?
No
Match Arbitration
Yes
RLPDQ Truth
Match Quality Metrics
Scoring
Dashboard

Fig. 3 – An RLPDQ Block Diagram

Fortunately most of these predicted positive matches between the two RL systems will agree because both systems are adept at doing record linkage, however, being different, they tend to make different mistakes. The matches that don't agree go to match arbitration to seek some additional positive matches. Both of these sets of positive matches then go into the Truth file. Using the Truth file you can take the sampled production data along with the metadata associated with it and do a scoring step. From that you can get quantitative quality metrics we have been discussing: the true positives and false positives, true negatives and false negatives.

In addition these data can be put onto a data quality dashboard (Ref. 1) for easy viewing by management to monitor system performance in near real time during production.

So the basic Record Linkage PDQ (RLPDQ) concept this: by using an independent record linkage system that has different characteristics and/or settings than the production record linkage system, one can use automation to help with this very difficult testing problem.

As you can see in Fig. 4 the challenge is to efficiently and cost-effectively get from *comparison space* that is of order N^2 to the neighborhood of the true positive matches that is of order N. So, for example, if the two files have roughly 1,000 records each, the number of comparisons the record linkage system has to perform to determine matches is about 1 million, whereas the number of final linked records will be around 1,000.

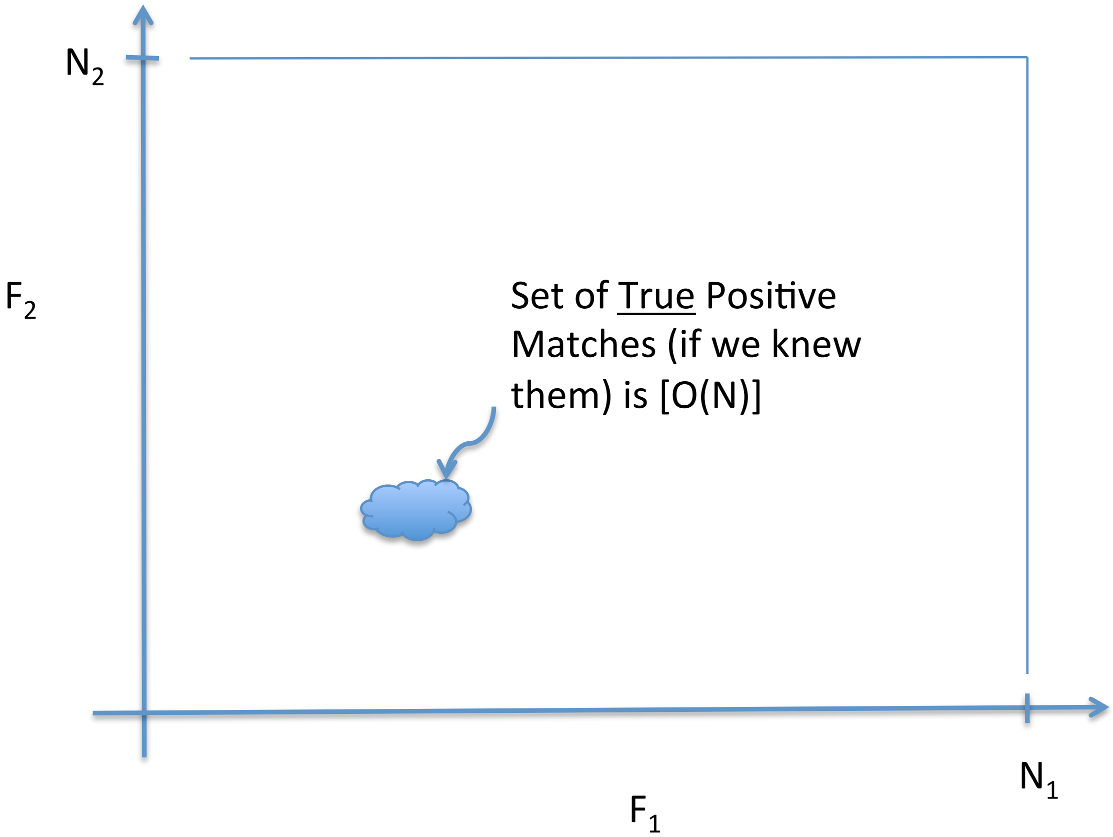


Fig. 4 – Comparison Space is of O (N^2); Matches are O (N)

How Does RLPDQ Work?

In order to give an idea of how RLPDQ works, we will now show a sequence of four set-theoretic diagrams. These diagrams are drawn in a chunk of *comparison space*, in which each point corresponds to a record pair, consisting of data about one *entity* from F_1 and another *entity* from F_2. An *entity* can generally be almost anything you wish, but for our purposes here just think of entities as people, or more specifically, heads of households. So the record linkage system's job is to examine all N^2 of these entity pairs, and decide if there is a positive match between them. We assume here that duplicates have been removed from the two files being linked for analytical simplicity (it's a good idea in practice also).

In Fig. 5 we show a cloud representing the positive matches predicted by the production RL system; this set of points in comparison space we label m_P. This is the production system's best shot at what it thinks the correct matches are, but keep in mind that whereas most of the matches may be true positive matches, some may be false positives. Also, there will likely be true positive matches outside of set m_P and some of these may be captured by the independent RL system.

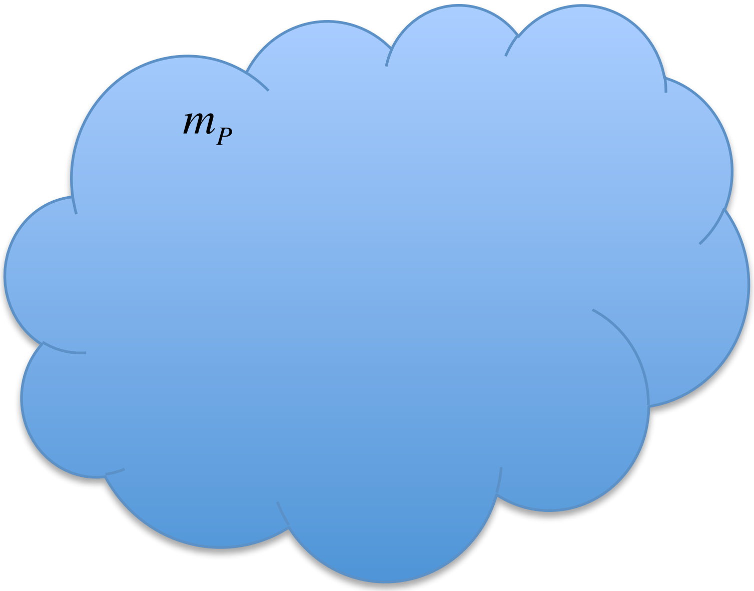


Fig. 5 – Production Record Linkage Predicted Matches (m_P)

In Fig. 6 we add in the predicted positive matches from the independent RL system, which we label m_I. The intersection of these two sets is also shown in Fig. 6, and is labeled $m_P \cap m_I$. This intersection represents the positive matches predicted by both the production RL system and the independent RL system, and is a first order estimate at positive matching Truth, automatically, and without any human analyst effort.

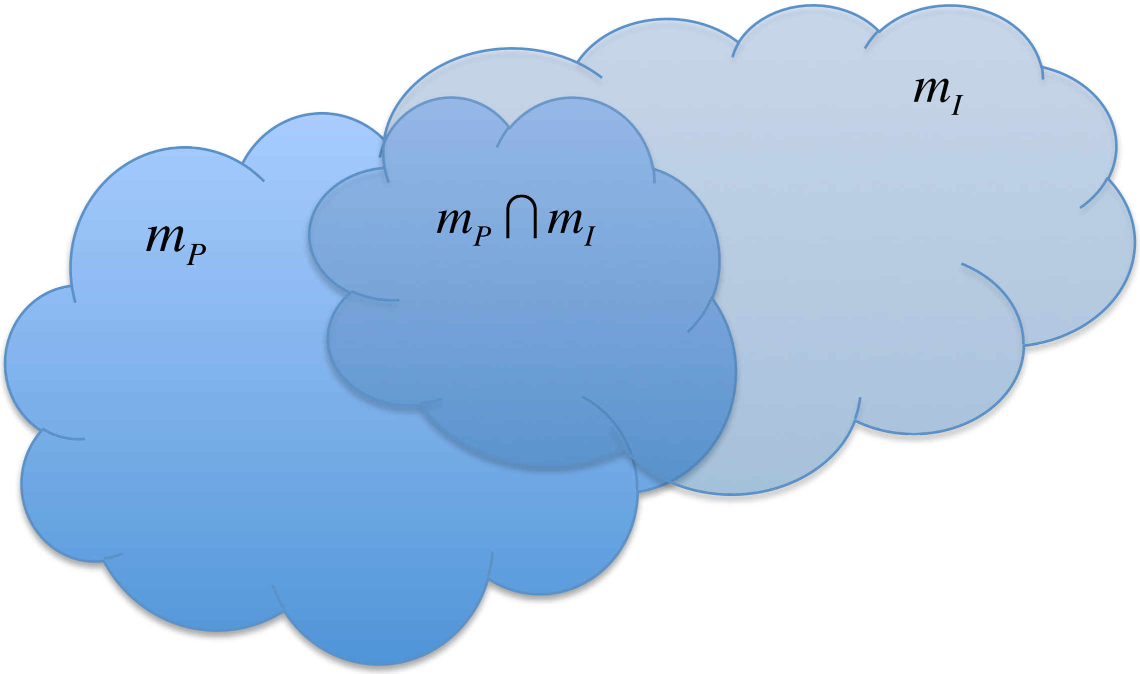


Fig. 6 – Add Independent Record Linkage Predicted Matches (m_I)

Another important aspect of Fig. 6 is the union of the sets m_P and m_I given by $m_P \cup m_I = m_P + m_I - m_P \cap m_I$. We define this space to be *entity-matching space*, which is where we expect to find almost all of the true positive matches. This space is a bit different than the entity space defined by Ref. 4.

The only true positive matches that would not be found in *entity-matching space* are true positive matches not found by <u>either</u> RL system. Although this is certainly possible, the number of these matches that could not be found by RLPDQ as described herein is expected to be very small compared to $m_P \cap m_I$, assuming both the production and the independent RL systems are production-quality and dissimilar.

Next in Fig. 7 we indicate the set m_U that represents the as yet "unmatched" portion of *entity-matching space* that should explored by arbitration to seek additional correct positive matches. This set is calculated as $m_U = m_P + m_I - 2m_P \cap m_I$.

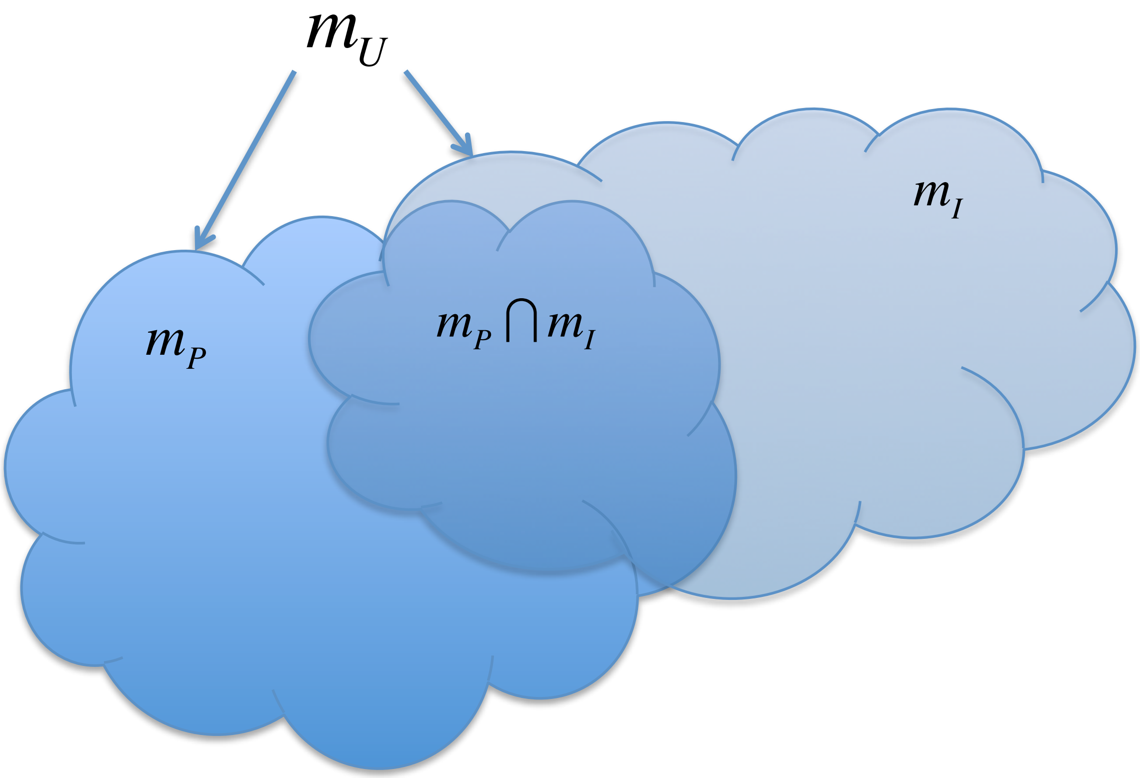


Fig. 7 – Seek Additional Positive Matches Predicted by One RL System but Not the Other in m_U

This arbitration process could be a semi-automated process using human analysts as was done in Ref. 1. A major advantage of RLPDQ is that upwards of 90% of the true positive matches may be found in the intersection $m_P \cap m_I$ that is completely determined by automation. Another advantage that can be nicely exploited in arbitration is that one RL system (or the other) is likely to predict a particular positive match correctly even if the two systems don't agree on that match.

Now, in Fig. 8, we show some additional true positive matches that could be found through the arbitration process, denoted by Δ_I and Δ_P. These matches are also false negative matches for the Independent and Production RL systems respectively.

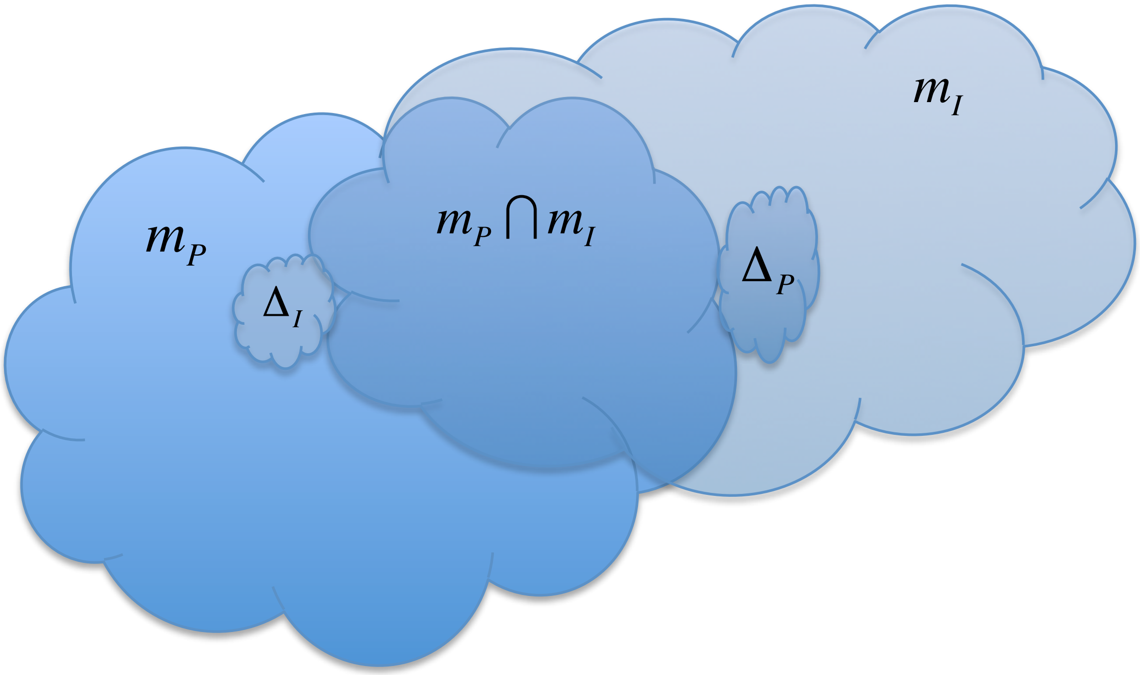


Fig. 8 – Find Some Additional Positive Matches (Δ_I & Δ_P)

We now know the quantities $N_1, N_2, |m_P|, |m_I|, |m_P \cap m_I|, |\Delta_P|, \& |\Delta_I|$, where the absolute value signs denote we are referring to the number of entity pairs in the set enclosed by these signs.

In order to start filling out a confusion matrix, we also need to decide what N should be, and this is not necessarily obvious given that *N₁* and *N₂* may not be the same values. Some have even decided to make $N = N_1 N_2$, which excessively complicates the analysis, as pointed out in Ref. 4. With no loss in generality for this application, we choose to define $N = \max(N_1, N_2)$.

At this point, our best estimate of the number of true positive matches is given by the equation $M = |m_P \cap m_I| + |\Delta_I| + |\Delta_P|$. We now know all the row and column <u>sums</u> for the Production RL system confusion matrix (and the Independent one as well which can sometimes be useful for deeper analysis).

In addition, we also know <u>all</u> the rest of the Production RL system confusion matrix elements. The number of true positives is $TP_P = |m_P \cap m_I| + |\Delta_I|$, the number of false positives is $FP_P = |m_P| - |m_P \cap m_I| - |\Delta_I|$, the precision is $c_P = TP_P / |m_P|$, the number of false negatives is $FN_P = |\Delta_P|$, and since by definition $N = TP_P + FP_P + FN_P + TN_P$, the number of true negatives is $TN_P = N - |m_P| - |\Delta_P|$.

Given that the elements of the confusion matrix for the production RL system is known, it is possible to perform a Receiver Operating Characteristic (ROC) analysis as outlined in Ref. 1 by computing the True Positive Rate (TPR) as $TPR_P = TP_P / M$ and the False Positive Rate (FPR) as $FPR_P = FP_P / (N - M)$. In addition, you can compute the overall Accuracy (ACC) as $ACC_P = [M \times TPR_P + (N - M) \times (1 - FPR_P)] / N$.

A Numerical Example

All this is admittedly rather theoretical, and so it is instructive to show how all this works with a numerical example. In Ref. 3, we showed how synthetic data could be useful for testing a census-like administrative records system that does record linkage with data from another agency to improve census data. In order to fully appreciate all the fine points, one should read Ref. 3. However, we will replicate the essence of that study here for convenience.

Using our Dynamic Data Generator™, in Ref. 3 we created two synthetic data sets of about a thousand records each; the first set (F_1) resembled census-type data, and the second (F_2) resembled tax-type data. Using our terminology from above, the actual number of entities (heads of households) in each file were N_1 = 985 and N_2 = 852. Using the larger number, we set N = 985 for subsequent confusion matrix-type analysis. Note the comparison space is much larger, given by N_1N_2 = 985 x 852 = 839,220.

In Ref. 3, we set up an experimental record linkage engine in two ways to simulate two different record linkage systems, E_1 (using five comparison fields) and E_2 (using four comparison fields). The portion of the experimental results needed here are shown in Fig. 9 below:

Systems ⇒ Data Type ⇓	RL System #1 (E_1)	RL System #2 (E_2)
Predicted Matches (m)	808	925
True Positives (TP)	805	818
False Positives (FP)	3	107
False Negatives (FN)	43	30
Precision (c)	0.9963	0.8843
Accuracy (ACC)	0.953	0.861

Fig. 9 – RL Results from Reference 1 for Two RL Systems

Clearly, and as expected in this relatively simple example, RL System #1 (E_1) was the better of the two, despite the fact that RL System #2 (E_2) predicted more positive matches; the problem being that the positive matches predicted by E_2 had many more false positives.

What we do now for this RLPDQ numerical example is play the game backwards; we define the RL System E_1 as our Production RL System and E_2 as our Independent RL System. Then we apply the above PDQ theory, step by step, to see how it comes out relative to the results already obtained from Ref. 3.

We will show some detailed, but helpful, graphs as we go, starting with Fig. 10, where we have placed the $|m_P| = 808$ predicted positive matches from the Production RL System (in blue) on a small portion of comparison space. Two points of explanation are in order: first, the grid in Fig. 10 is only 50 x 40 = 2,000 record pairs, which is over 400 times smaller than the entire comparison space; and second, the actual location of the specific record pairs in comparison space is arbitrary, as one could freely interchange rows and interchange columns to group them as desired.

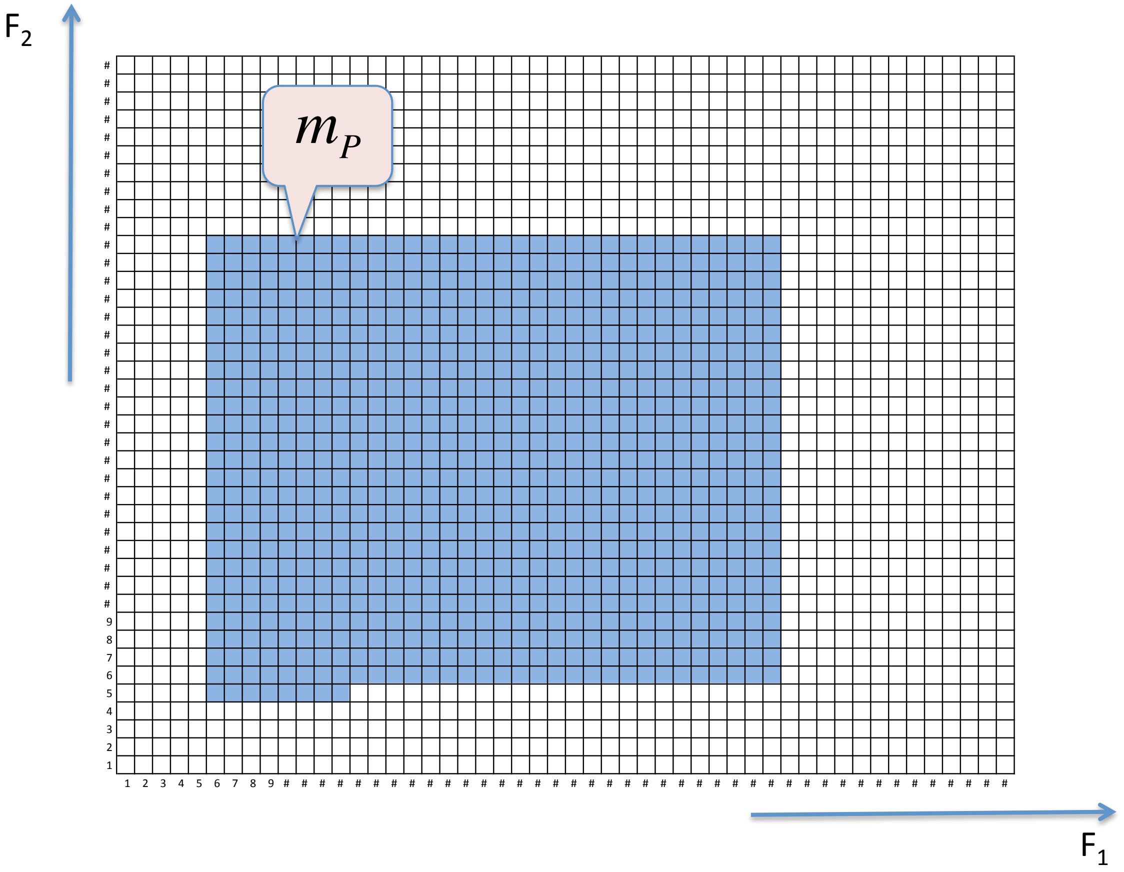


Fig. 10 – Production Predicted Positive Matches (m_P); the size of m_P is $|m_P| = 808$ shown here on a small (50x40) portion of Comparison Space

A proper RL system not only tells you how many positive matches are predicted, it also tells you which record pairs they are. By sorting, then, it is possible to determine which record pairs are in common between E₁ and E₂, and plot them on comparison space as shown in Fig. 11.

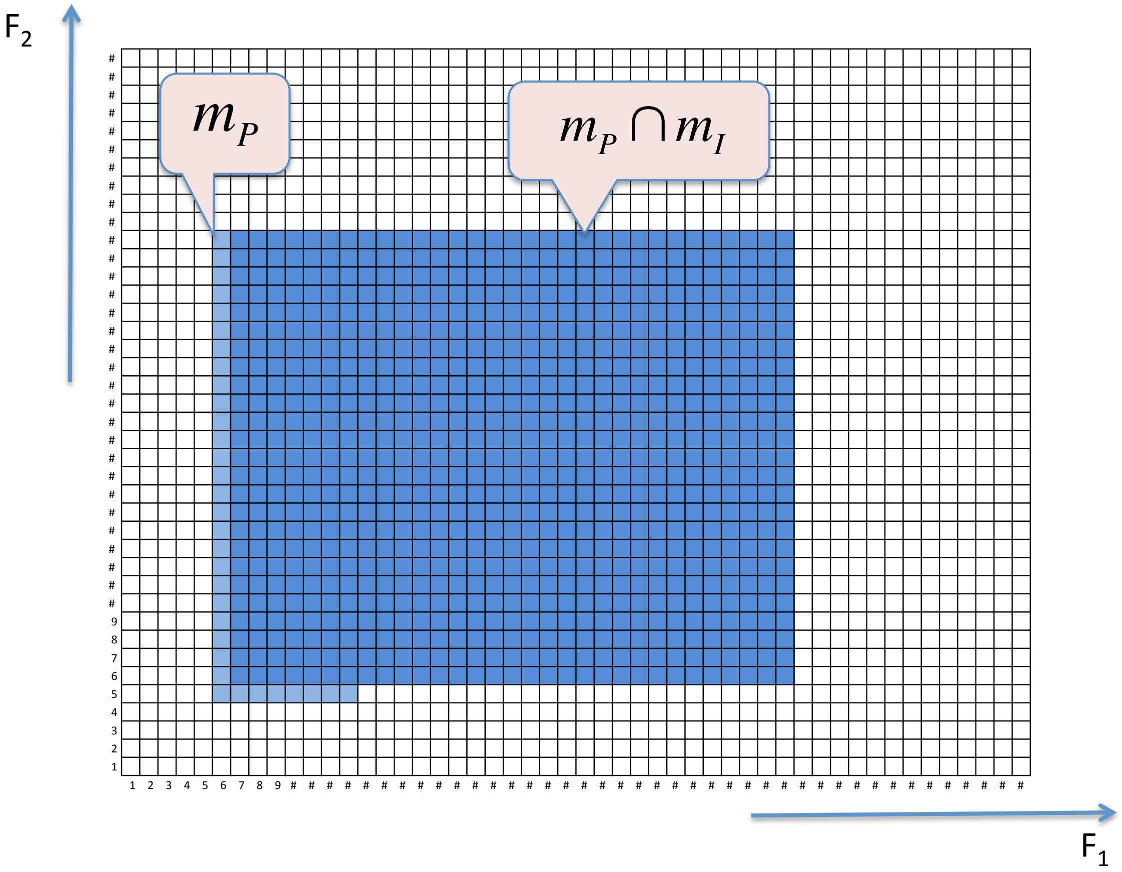


Fig. 11 – Add Predicted Positive Matches in Common Between Production and Independent RL Systems ($m_P \cap m_I$)

You could count them from Fig. 11, but the actual size of the set $m_P \cap m_I$ is given by $|m_P \cap m_I| = 775$.

Continuing our build-up of detailed set diagrams, we now add in the independent predicted positive matches (m_I), as shown in Fig. 12. Now, $|m_I| = 925$, so most of the set m_I is hidden under the set of common matches $m_P \cap m_I$; but, of course, that is a good thing!

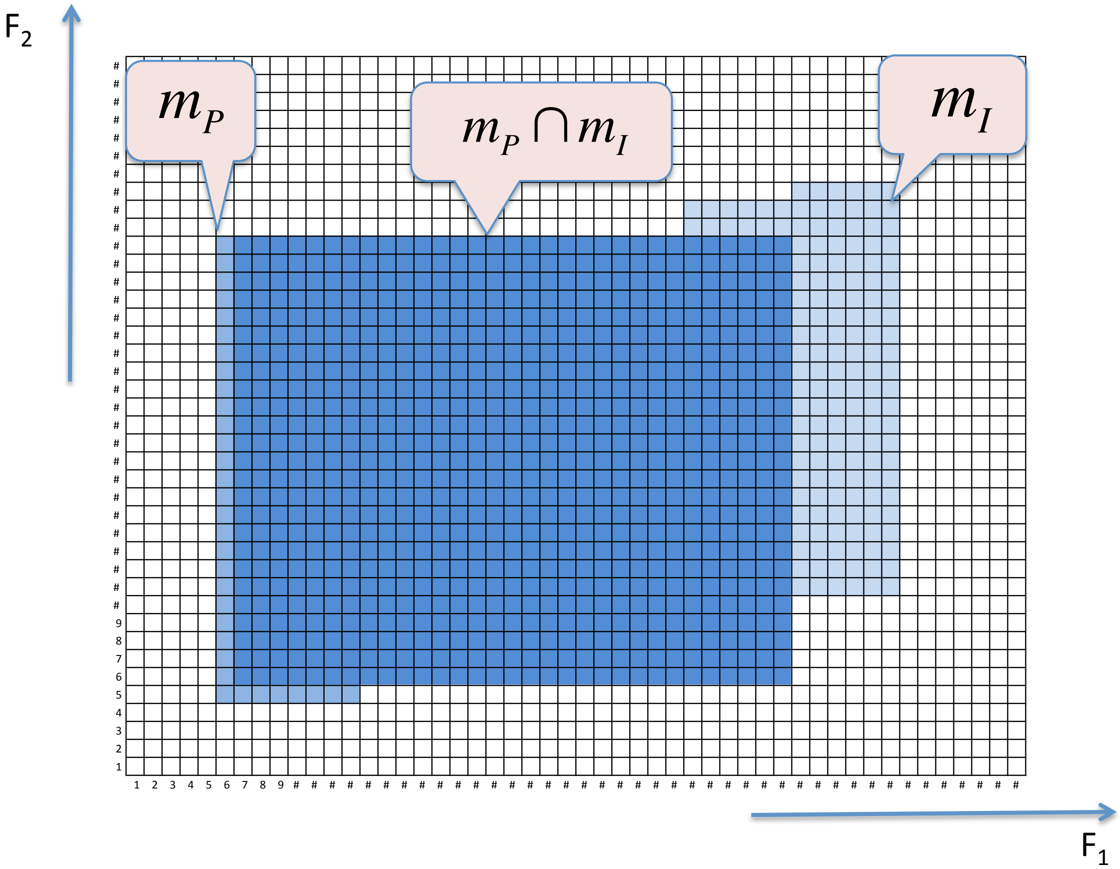


Fig. 12 – Add Independent Predicted Positive Matches (m_I); the only part of m_I that shows here is the part not in common with Production)

Starting our arbitration process, (by manually looking in that portion of m_I that is outside the intersection $m_P \cap m_I$), we readily found the additional production false negatives, denoted by Δ_P, and added them into Fig. 13.

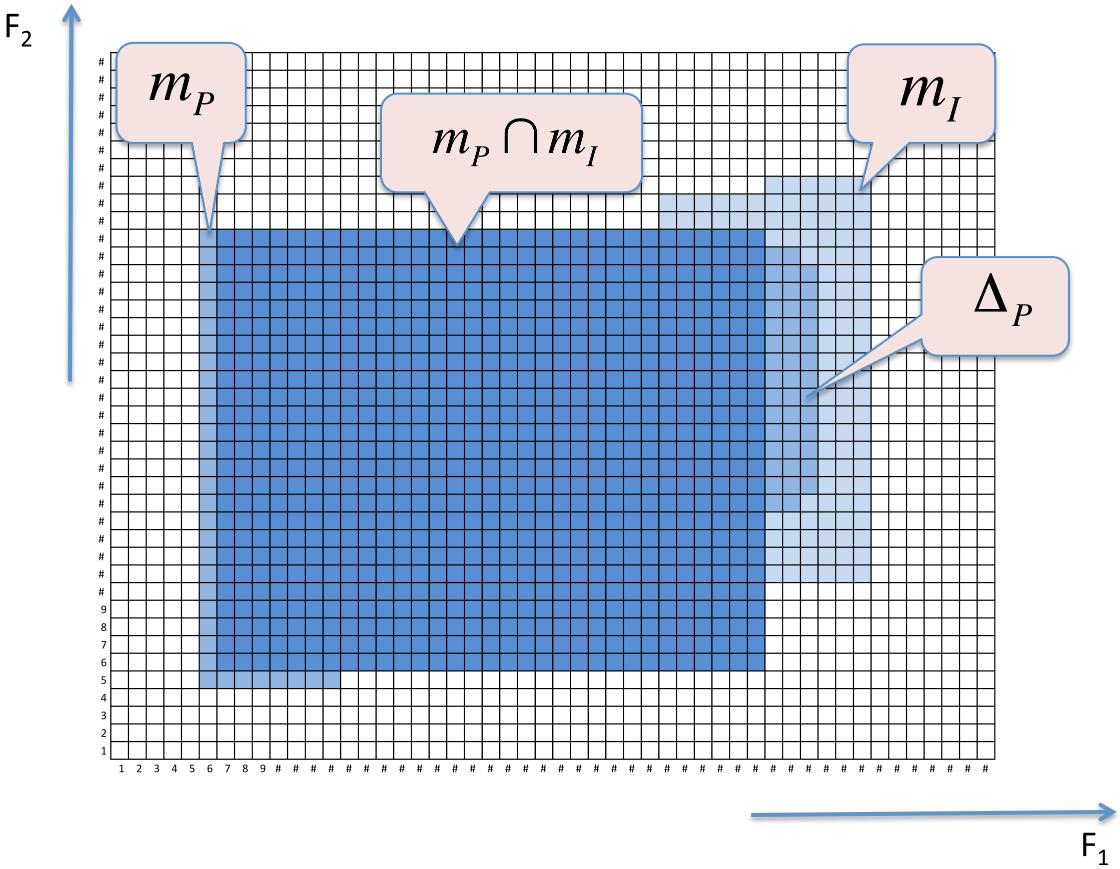


Fig. 13 – Add Production False Negatives (Δ_P)

The size of the set Δ_P is $|\Delta_P| = 43$, and represents the additional true positive matches that were found by the independent RL system, but not found by the production RL system.

Conversely, we look into the production RL system's predicted positive matches not in common with the independent RL system, and find the independent RL system's false negatives, denoted by Δ_I and shown in Fig. 14.

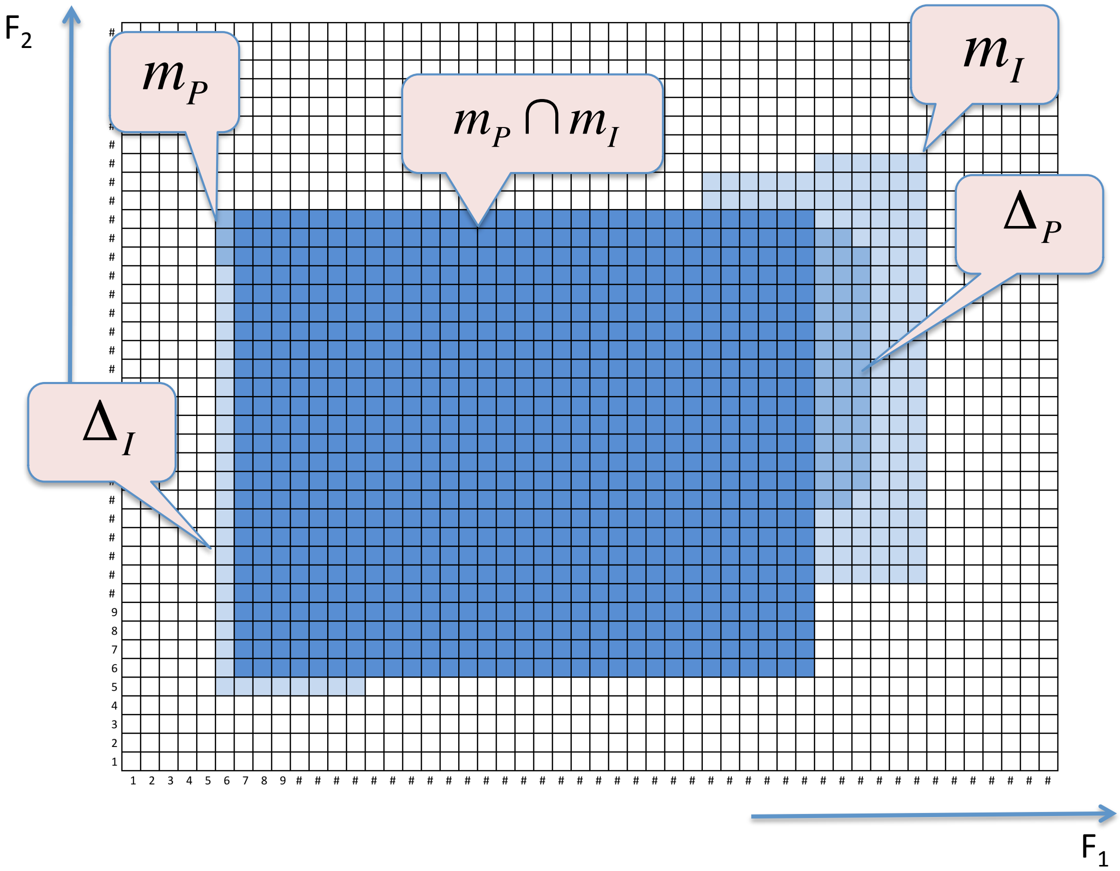


Fig. 14 – Add Independent False Negatives (Δ_I)

The size of the set Δ_I is $|\Delta_I| = 30$, and represents the additional true positive matches found by the production RL system but not found by the independent RL system.

Note that the entire (unmatched) space that had to be examined for arbitration was $m_U = m_P + m_I - 2m_P \cap m_I$; the size of this unmatched space is only $|m_U| = 808 + 925 - 2(775) = 183$. This is merely 183/839,220 = 0.000218, or 0.022% of comparison space! That's why manual inspection worked here; for larger files, a semi-automated graphic terminal would be needed as done in Ref. 1.

Perhaps it has been a bit tedious, but we have now built up, piece-by-piece, the final representation of entity-matching space, $m_P \cup m_I = m_P + m_I - m_P \cap m_I$, as shown in Fig. 15. The size of entity-matching space is $|m_P \cup m_I| = 808 + 925 - 775 = 958$.

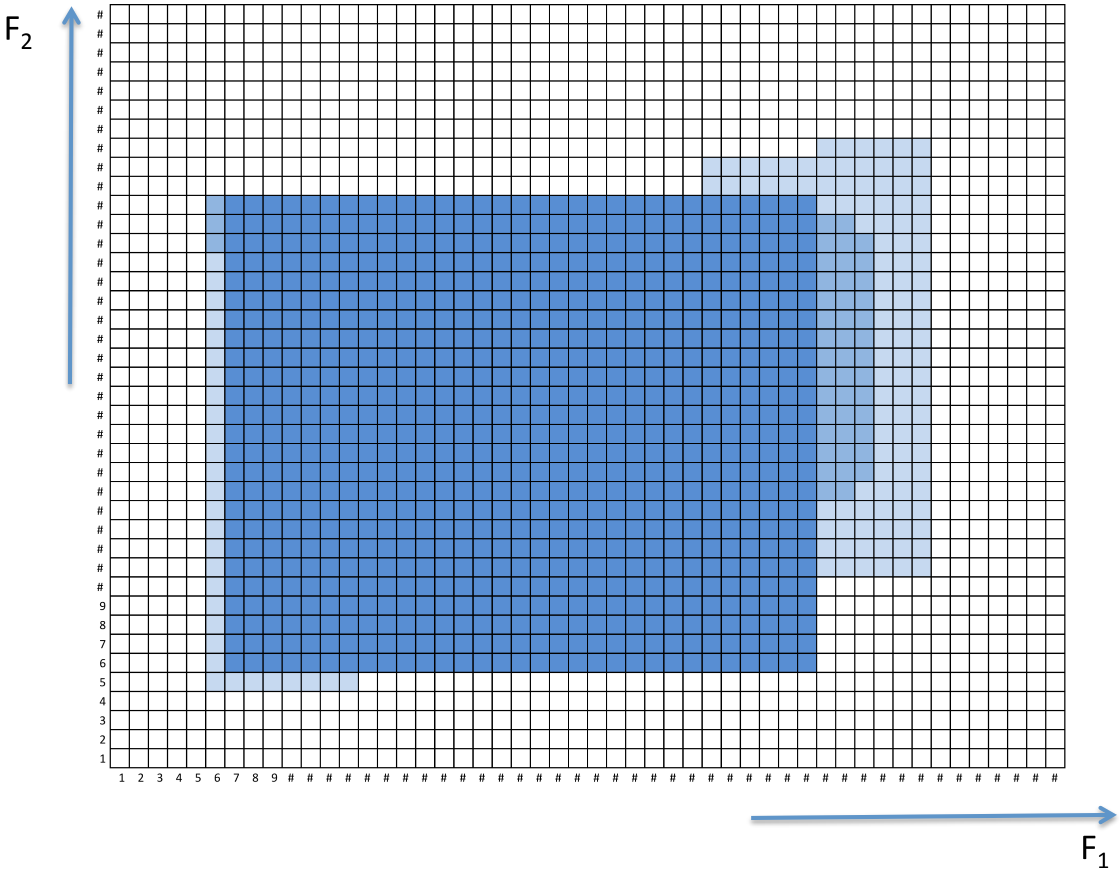


Fig. 15 – Final Entity-Matching Space ($m_P \cup m_I$)

As already mentioned above, we have intentionally plotted these last six figures on a small portion of comparison space shown as a grid. In order to more clearly see how small entity-matching space is relative to the entire comparison space, look at Fig. 16.

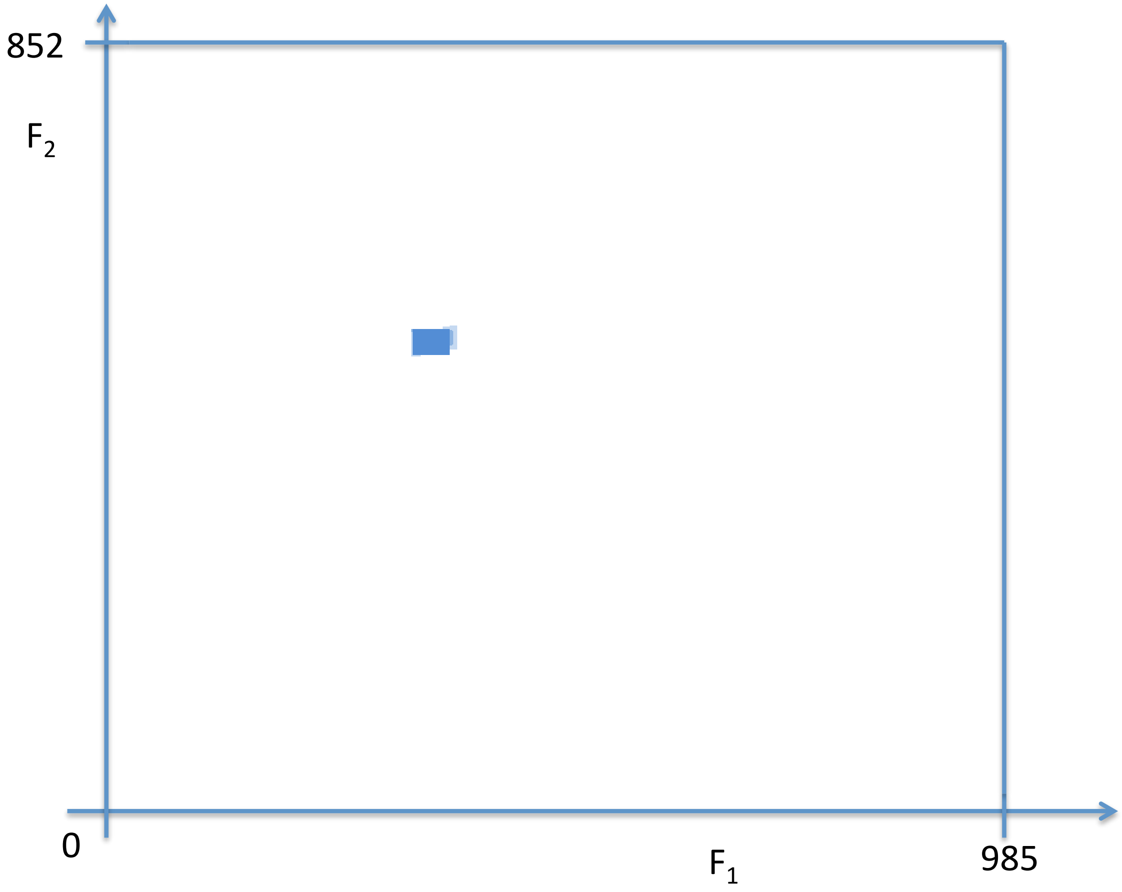


Fig. 16 – Entity-Matching Space (Bluish) Embedded in the Full Comparison Space (to scale)

The message here is what RLPDQ does for you is use automation to get from the huge rectangle (comparison space) down to the (876 times smaller) bluish area (entity-matching space), and the bulk of entity-matching space is the common true positive matches found automatically. In the numerical example just given, over 90% of the true positive matches were found automatically in the matches in common between the two RL systems.

To complete this numerical example, we have our estimate of the size of the true positive matches $M = |m_P \cap m_I| + |\Delta_I| + |\Delta_P|$ = 775 + 30 + 43 = 848, which is exactly (in this case) the correct number of true positive matches from Ref 3. If there were some true positive matches outside entity-matching space, this would not (generally) be the case.

You can calculate using the equations given above and the numerical results just given all the rest of the production confusion matrix elements and the ancillary Receiver Operating Characteristic (ROC) results, which are all in perfect agreement with the actual results found in Ref. 3. Specifically, we get:

True Positives: $TP_P = |m_P \cap m_I| + |\Delta_I|$ = 775 + 30 = 805

False Positives: $FP_P = |m_P| - |m_P \cap m_I| - |\Delta_I|$ =808 – 775 – 30 = 3

Precision: $c_P = TP_P / |m_P|$ = 805/808 = 0.9963

False Negatives: $FN_P = |\Delta_P|$ = 43

True Negatives: $TN_P = N - |m_P| - |\Delta_P|$ = 985 – 808 – 43 = 134

True Positive Rate: $TPR_P = TP_P / M$ = 805/848 = 0.949

False Positive Rate: $FPR_P = FP_P / (N - M)$ = 3/(985 – 848) = 0.022

Accuracy: $ACC_P = [M \times TPR_P + (N - M) \times (1 - FPR_P)] / N$
= [848 (0.949) + 137 (0.978)]/985 = 0.953

Conclusions

1. Just as was done for forms processing in Ref. 1, the PDQ approach may be used to cost-effectively and precisely determine the "Truth" of production record linkage results to create quality metrics or to aid in training.

2. These metrics can also assist in more rapidly finding opportunities for record linkage algorithm improvement by clearly pointing out pockets of error, particularly when using a data quality dashboard as was done in Ref. 1.

Future Work

The next step is to put RLPDQ to work on a larger actual record linkage problem, and to begin to develop some bounds for precision of the quality metrics.

References

1. Paxton, K. Bradley, Spiwak, Steven P., Huang, Douglass, and McGarity, James K., *Testing Production Data Capture Quality*, Proceedings, Federal Committee on Statistical Methodology (FCSM), Washington, DC, (2012)

2. Winkler, William E., *Automatically Estimating Record Linkage False Match Rates*, U.S. Census Bureau, Statistical Research Division Research Report, rrs2007-5, (2007)

3. Paxton, K. Bradley, and Hager, Thomas, *Use of Synthetic Data in Testing Administrative Records Systems*, Proceedings, Federal Committee on Statistical Methodology (FCSM), Washington, DC, (2012)

4. Christen, Peter, and Goiser, Karl, *Quality and Complexity Measures for Data Linkage and Deduplication*, Studies in Computational Intelligence **43**, 127 – 151 (2007)

Contact:
K. Bradley Paxton
CEO, ADI, LLC
brad.paxton@adillc.net