G-LINK: A Probabilistic Record Linkage System

Antoine Chevrette, Statistics Canada

Abstract

At Statistics Canada, matching data without unique identifiers is a common practice. The probabilistic record linkage method developed by Ivan Fellegi and Allan Sunter[1] is the primary method recommended by Statistics Canada for this type of matching.

In recent decades, work began to generalize the Fellegi–Sunter algorithm in order to offer our community the opportunity to use this methodology within a computer application. The most recent version of this application is called G-LINK and is part of Statistics Canada's package of generalized systems.

By definition, a generalized system must be user-friendly, robust, fast, highly flexible and responsive to user demands. It will be interesting to discover from reading this article how it was possible to meet these criteria using the latest user interface and development technologies.

Introduction

To fully meet the challenges involved in making the transition from mathematical theory to computer application, it is first necessary to lay the theoretical groundwork. To do this, a general review of the Fellegi–Sunter algorithm will be provided to show the algorithm in relation to its computer avatar G-LINK.

Certain critical methodological elements pose a challenge with respect to both technology and the user interface, and these will be examined in detail. The solution used in G-LINK for each of those challenges will also be described in detail.

Finally, there will be a discussion of future computational developments in G-LINK to carry it to even greater heights.

Methodological review

What is probabilistic linkage? This is a linkage with no unique identifier, for which we estimate the likelihood that the records correspond to the same entity. From this definition, it can easily be inferred that there is no point in applying probabilistic linkage to data that have a unique identifier (e.g. a social insurance number); this type of matching can easily be done using direct linkage.

The purpose of G-LINK lies in its ability to provide both internal and external probabilistic linkage. Internal linkage is used to find groups of records that refer to the same entity within the same file, while external linkage is used to find groups of records in two different files referring to the same entity.

Probabilistic record linkage generally consists of six separate steps:

1. Generate potential pairs using a selection criterion
2. Generate rules and apply them to the potential pairs to derive probability ratios
3. Assign a status to the pairs using the probability ratios
4. Apply frequency probabilities (weighting factors)
5. Form groups
6. Resolve conflicts using mapping.

Generate potential pairs

The sole purpose of this step is to limit the number of potential pairs generated. In a perfect world, it would be essential to consider the set of all possible pairs. For example, in a linkage of two files, if the first file

contains 50,000 records and the second contains 20,000 records, then by taking the Cartesian product of these two files we obtain $20{,}000 \times 50{,}000 = 1{,}000{,}000{,}000$ possible pairs to evaluate.

Unfortunately, since this is not a perfect world, and since the computers available to us cannot yet be said to have Herculean power, it is unimaginable to generate all possible pairs resulting from a Cartesian product of files of even modest size. Just imagine a Cartesian product of 30 million times 30 million.

To get around this problem, it is crucial to use the first step of the algorithm, which is to generate potential pairs using a selection criterion based on direct mapping. A user might decide to generate all potential pairs resulting from mapping in which the first three characters of the postal code match and the sex is the same.

Generate and apply rules

How do we determine whether a pair corresponds to the same entity? To do this, we need to be able to measure the level of agreement in the information characterizing the pairs. This level of agreement is actually a probability ratio that is assigned to the pair by means of rules and levels of agreement.

Predefined character, numerical and date-type rules exist within G-LINK. A rule can be constructed on the basis of several types of comparisons called levels of agreement.

To clarify the concept of rules, there is nothing better than a concrete example.

TABLE A			TABLE B		
Surname	**Given Name**	**Birth Year**	**Surname**	**Given Name**	**Birth Year**
SMITH	SUSAN	1940	SMITH	S	1939

(Table 1)

To determine whether the records in Table A and Table B represent the same person (Table 1), it is advantageous to use different rules. For this example, use of a character-type rule on the given name and surname and a date-type rule on the year of birth will serve the purpose.

Rule	**Outcome level**		**Result**	**Comparison**
Surname	1	Complete match	Agreement	SMITH=SMITH
	2	Partial match (Nysiis)	Not evaluated because the preceding outcome level was true	NA
Given Name	1	Complete match	Disagreement	SUSAN ≠ S
	2	Partial match (first character)	Agreement	S=S
Birth Year	1	Complete match	Disagreement	1940 ≠ 1939
	2	Partial match (Year minus one)	Agreement	1940-1 = 1939

(Table 2)

From Table 2, it is easy to see that the first level of the Surname rule is in agreement and that the second level of the Given Name rule and the Birth Year rule are also in agreement.

An outcome level can yield only three different values: an agreement, a disagreement or a missing (value missing on one or both of the sides compared). When a probability is associated with an outcome level, its probability ratio can be calculated.

How do we assign a probability? It is first necessary to give the probability that the outcome level is true when the pair belongs to the set of related pairs (and is thus a good pair). It is also necessary to give the probability that the outcome level is true knowing that the pair belongs to the set of unrelated pairs (and is thus a bad pair).

Using the rules in Table 2, we have

Outcome level probabilities												
Rule:	Surname				Given Name				Birth Year			
	1	2	M	D	1	2	M	D	1	2	M	D
Linked sets	.70	.20	.05	.05	.80	.10	.05	.05	.87	.08	.02	.03
Non-linked sets	.01	.04	.05	.90	.02	.05	.05	.88	.01	.06	.02	.91
Prob. ratio	70.00	5.00	1.00	.06	40.00	2.00	1.00	.06	87.00	1.33	1.00	.03

(Table 3)

For example, level 1 of the Surname rule (complete comparison) is true 70% of the time when the pair belongs to the set of linked pairs, while it is true 1% of the time when the pair belongs to the set of non-linked pairs. The probability ratio is calculated by dividing the probability for the linked set by the probability for the non-linked sets. Below is the mathematical formula describing the probability that the records will be linked for a particular outcome level (probability ratio):

$$PR_i[r_i(a,b)] = \frac{P(r_i(a,b)|(a,b) \in L)}{P(r_i(a,b)|(a,b) \in N)}$$

It should be noted that the higher the probability ratio, the greater the probability of having a linked pair; the inverse is also true.

To find the probability ratio of the pair and not of the rule, we need only multiply together the probability ratios of the rules. However, to be valid, this multiplication requires that the rules used be independent.

$$PR(R) = PR(r_1) \times PR(r_2) \times ...PR(r_n)$$

Pair						
	Rule					
	Surname		Given Name		Birth Year	
Outcome level	C (Agreement)		PA (Partial agreement)		PA (Partial agreement)	
Probabilities						
Linked set	0.7		0.1		0.08	
Non-linked set	0.01		0.05		0.06	
Probability ratio	70.00		2.00		1.33	PR: 186.67

(Table 4)

In our example, the probability ratio for the pair is 186.67. Thus, this pair has 186.7 times the chance of being in the set of linked pairs than of being in the set of non-linked pairs.

Assign a status to pairs

The status of a pair is a value that is assigned to it in order to categorize it. A pair can be categorized as definitive (good pair), possible (to be considered) and rejected (not considered). There is also a fourth status, "excluded," which is used in applying the rules. If in that process, the probability ratio fails to reach a certain value (called the cut-off threshold), the pair is excluded and becomes isolated from the rest of the process.

In order to assign a status to pairs, thresholds are needed. A lower threshold and an upper threshold are used:

- T_i Lower threshold
- T_s Upper threshold

The pairs are initialized as follows:

- Weight(a,b) < T_i Status = R (Rejected)
- T_i <= Weight(a,b) < T_s Status = P (Possible)
- Weight(a,b) >= T_s Status = D (Definitive)

Apply frequency probabilities

To refine the probability ratios, frequency weights can be applied for all outcome levels of a rule that has a result. A non-linked frequency weight will replace the non-linked portion of the outcome level weight. A linked frequency weight will replace the linked portion of the outcome level weight. Finally, a frequency weight having both linked and non-linked portions will replace the corresponding two components of the outcome level weight.

Non-linked frequency weights can be calculated using the input tables (Table A and Table B). For example, say that we have agreement on the surname Smith and agreement on the surname Aardvark. Since Smith is a much more common value than Aardvark, it is more useful to refine the weights by assigning more importance to agreement on Aardvark (a rarer value). Below is the formula used to calculate this type of frequency weight:

$$NFreqWeight_i = -10 * Log_2\left[\frac{Numb\,\mathrm{Re}\,cs(table_j)}{TotalNumbNonmissing\,\mathrm{Re}\,cs(table_j)}\right]$$

The frequency weights can also be calculated based on the outcome levels (linked frequency weights). For example, say that an outcome level uses the similarity algorithm (developed by William Winkler) returning as a value a percentage of exactness between two strings of characters. More weight should be assigned to rare values (99%) and less weight to common values (90%). The formula used to calculate this type of weight is similar to the previous formula:

$$LFreqWeight_i = 10 * \log_2\left[\frac{NumbLinkedPairs_i}{NumbLinkedPairs}\right]$$

It is also possible to calculate and apply frequency weights based on the outcome level results but also on the non-linked portion of the pairs. This non-linked portion, commonly known as the random portion, is constructed by randomly selecting pairs from Table A and Table B.

$$NFreqWeight_i = -10 * \log_2\left[\frac{NumbNonlinkedPairs_i}{NumbNonlinkedPairs}\right]$$

Form groups

Why is it necessary to form groups within pairs? The answer is simple: a number of pairs may be inter-related among themselves. An example of this is where record 1 in Table A is related to record 4 in Table B, but record 4 in Table B is also related to record 10 in Table A, etc.

More specifically, groups are generated from pairs according to their status. There are two types of groups: weak groups and strong groups. Weak groups are made up of records with possible and definitive links. Strong groups are internal to weak groups and contain only records linked together by definitive links.

In summary, this step in the process serves to organize pairs in such a way as to be able to easily resolve conflicts by mapping.

Resolve conflicts using mapping

Conflict resolution is the last step before exporting the linkage results. It can happen that the returned pairs referring to the same entity are multiple. Mapping is used to specify the type of results sought in record matching. There are four types of mapping: multiple to multiple (output of the creation of groups), one to multiple, multiple to one and one to one.

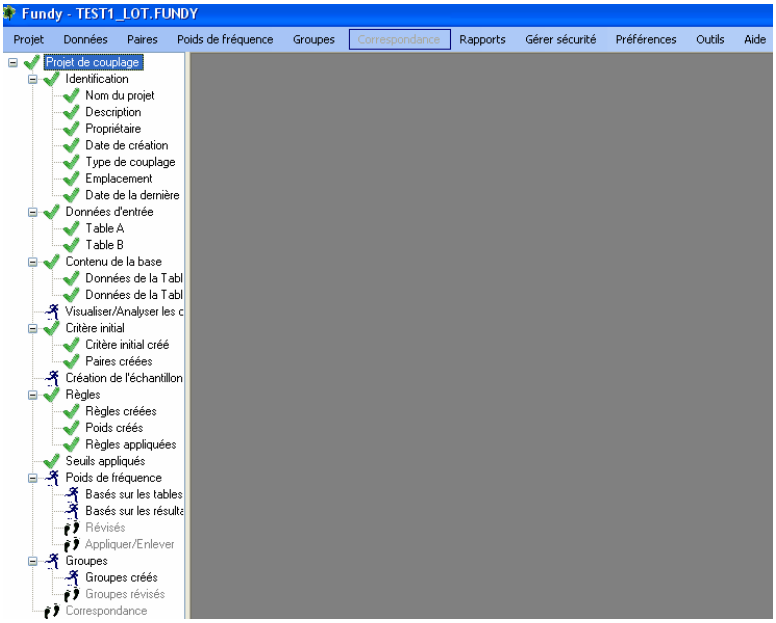
Challenges

The above brief review of the methodology will make it easier to understand the challenges that arose in constructing the computer avatar of the Fellegi–Sunter algorithm. Don't worry; no advanced knowledge of computer science is required in order to understand the following sections.

Synthesizing the information (using a status tree)

A generalized system must be clear and user-friendly; above all, it must not cause the user to feel confused. The methodology review presented the six main steps in the algorithm used, and it presented them in a specific order. That order is not required when creating and implementing a project using G-LINK. For example, a user could first calculate the frequency weights based on the tables (non-linked), then formulate the rules and create potential pairs. Moreover, at any stage in the project, a user could decide to re-apply the previous steps. Why does G-LINK offer this flexibility? Because as a generalized system, it must meet the needs of all its users, and depending on the project, the order of operations may vary.

What is a clear and straightforward way to present the information on the stages of a project? By showing the steps in tree form. The use of a tree (Figure 1) provides users with a visual cue enabling them to determine instantly what stage their project is at and to determine everything that they can or cannot do. This way of presenting the information enables even novice users to intuitively know and apply the steps involved in record linkage.



(Figure 1)

The tree is composed of nodes, with each node representing a step in the algorithm. The nodes can be expanded to show the sub-steps. Each node has an image attached to it. There are three images: a check mark indicating that the step has been completed; a runner, indicating that the step can be carried out; and a pair of feet, indicating that the step cannot be carried out at this time (since it depends on a later step).

The internal architecture of the system lends itself to possible changes to paths and allows the programmers of the software to add nodes. If a step had to be added in the future, this would have only a minor impact on implementation.

Accessing the data

Accessing data outside the G-LINK system presented the development team with two completely different challenges: importing the data and displaying them.

If users cannot import the data to compare in G-LINK, they will be unable to use the software, and this is unacceptable for a generalized system. Since the majority of our users use SAS, one of the importation priorities is the importing of SAS files. So how can SAS data be imported in conjunction with Visual Basic.net (the technology used to develop the software interfaces)?

By chance, my previous work experience has led me to examine this quite interesting question. If you turn to the article 'SAS® Integration Technologies, UNIX and Visual Basic .Net Integration Procedure,'[2] you will find the solution to the problem of combining the two technologies. In brief, the interface (Visual Basic .net) uses the built-in features of SAS to create an SAS session invisible to the user to access import data. It should be noted that to use this option, SAS must be installed on the user's computer. G-LINK also offers the possibility of importing data in the format of a space-delimited text file (flat).

After data from an SAS file or a flat file are imported, the user's first impulse will be to check that they are the right data. G-LINK therefore provides a tool for viewing the data. This simple idea may seem trivial, but it isn't.

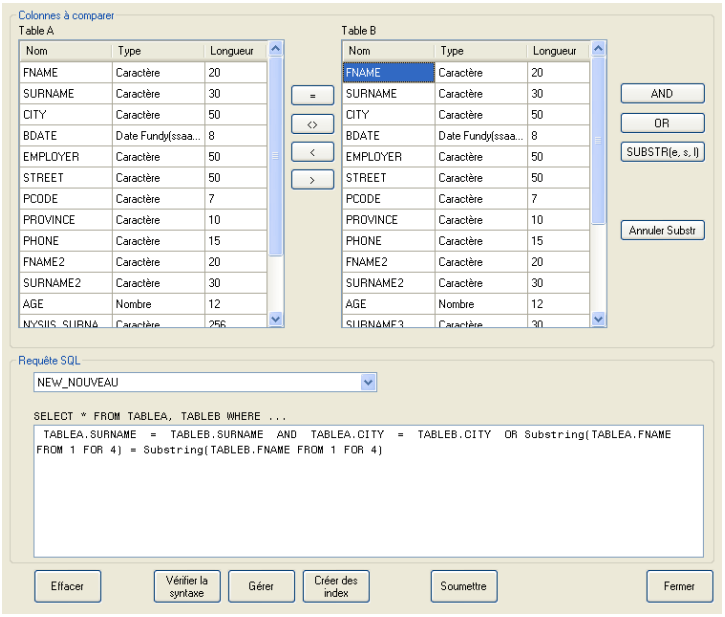
How can a table containing millions of records be displayed on the screen? The first attempt was simply to load the table into memory, and the result was a monumental failure caused by a lack of memory in the systems. A voluminous table can extend over several GB of spaces, and this often exceeds the total memory available in our operating systems. So what is the solution to this problem?

As it happens, Microsoft has developed a technique called Just In Time Data Loading, whereby only the records that are to be displayed on the screen are kept in memory. The scroll bar of the active window directly controls the data refresh process – for example, 100 records at a time. Obviously, loading 100 records into memory is much more stable than loading millions of records!

Query to generate initial pairs with indexation

The task of generating the pairs is automated, but generating the query is not. The pleasure of constructing this query falls to the user, and a query to generate pairs can range anywhere from a trivial equality to a complex, multi-level query.

Since G-LINK is a generalized system, one of its main features is user-friendliness. The interface for generating the pairs query was built to meet this criterion. Accordingly, a user with no knowledge of PL-SQL can generate the PL-SLQ query with a few mouse clicks. The interface was built (Figure 6) to allow automatic selection of the operators and fields used for comparison. The interface also allows automatic generation of sub-strings. However, is the user knows the P-SQL language, he or she can simply edit or generate a query in the box designed for this purpose.



Speeding up the process: divide and conquer

It quickly became clear that this process was inefficient for large-scale projects. The “normal” architecture used to create modest- to medium-sized projects cannot be applied to large projects. To give an idea of size, a project is considered “large” when it involves millions of records and pairs. For example, a project that has 20 million records for Table A and Table B and generates some 300 million possible pairs is considered large. The G-LINK team therefore looked into the problem and found a solution to this complex challenge. The old saying “divide and conquer” was the perfect approach to developing that solution.

Normally, all the data are imported into a Table A and a Table B. The more information there is in a table, the longer the time required for search and comparison. It was therefore decided to give the user the choice of separating the input data into a number of sub-tables. Of course, the task of separating a table is automatically taken on by the system; the user need only tell the system how much data a sub-table should contain.

Potential pairs are then generated using the sub-tables, and the rules are applied to these potential pairs. This technique eliminates the bother of managing voluminous tables.

For example, say that we have a Table A containing 20 million records and a Table B containing 4 million records. If it is decided to separate the tables into sub-tables of 250,000 records, there will be 80 sub-tables representing Table A and 16 sub-tables representing Table B. The process of generating pairs and applying the rules will have to be repeated 480 times (80*16).

When Table A and Table B are separated into a number of sub-tables, an element of table independence is introduced, allowing the processes of pair generation and rule application to take place simultaneously in the different sub-tables. Thus, if G-LINK is executed on a server with a number of CPUs, several table combinations can be evaluated at the same time. For example, if eight CPUs are used, the system can execute eight pair generation/rule application processes simultaneously. Processing time will be reduced eightfold.

Finally, the user can choose not to retain excluded pairs (pairs that do not reach a predetermined threshold). This means that the application has the potential not to keep superfluous information and therefore it can, once again, avoid having to manage voluminous tables.

Conclusion

This article shows how the world of methodology and the world of computers have been brought together in the G-LINK system. Merely by reading this document, one can gain a good understanding of the methodology and the system. As you have seen, simple methodological ideas can become formidable challenges when it comes to implementing them on a computer.

Currently, G-LINK reproduces the methodology of the Fellegi-Sunter algorithm. Over time, however, other highly useful features will be added to the application.

Batch processing is an important feature that should become available in March 2011. This will make it possible to execute G-LINK from a command line by providing it with an XML configuration file. Accordingly, it will be possible to create and execute a new project without the assistance of the G-LINK interface. This feature will be greatly appreciated in a production environment where tasks are repetitive. Also, since G-LINK will be executed from a command line, it can be used from an SAS environment or any environment that can execute system (DOS) commands.

In the near future, it will be possible to request matching through a web service. A user will be able to submit a query for a specific record and do the matching of it on a complete table.

G-LINK provides an effective solution to complex record-matching problems. The very nature of our work here at Statistics Canada makes G-LINK an indispensable tool. The planned improvements will help make it even more valuable.

---

[1] Fellegi, Ivan; Sunter, Alan (December 1969). “A Theory for Record Linkage”.
Journal of the American Statistical Association 64 (328): pp. 1183–1210.

[2] Paper 011-2008, SAS® Integration Technologies, UNIX and Visual Basic .Net,
Integration Procedure Antoine Chevrette, Statistics Canada, Ottawa, ON