A comparison of probabilistic record linkage techniques in the Institute of Statistics of Andalusia

Pino-Mejías, Rafael
University of Seville, Department of Statistics
C/ Tarfia s/N
41012 Seville, Spain
E-mail:rafaelp@us.es

Cubiles-de-la-Vega, María-Dolores
University of Seville, Department of Statistics
C/ Tarfia s/N
41012 Seville, Spain
E-mail:cubiles@us.es

Caballero-Ruiz, Elisa
Institute of Statistics of Andalusia
Leonardo Da Vinci, nº 21,41071 Seville, Spain
E-mail:elisa.caballero.ext@juntadeandalucia.es

1. Record Linkage Process.

The glossary of statistical terms of OECD (Handbook of Vital Statistics Systems and Methods, 1991) says that Record Linkage (RL) refers to a merging that brings together information from two or more sources of data with the object of consolidating facts concerning an individual or an event that are not available in any separate record. RL arises from the need of many public and private organizations to identify duplicate records in a database or to match records in different databases but related to the same unit. Record linkage of files (Fellegi and Sunter 1969) is used to identify duplicates when unique identifiers are unavailable. It relies primarily on matching of names, addresses, and other fields that are typically not unique identifiers of entities. Traditionally, health sector and statistical agencies are the main users of these techniques, but data mining projects can involve large databases from various sources, and therefore RL is a tool improving the quality of data.

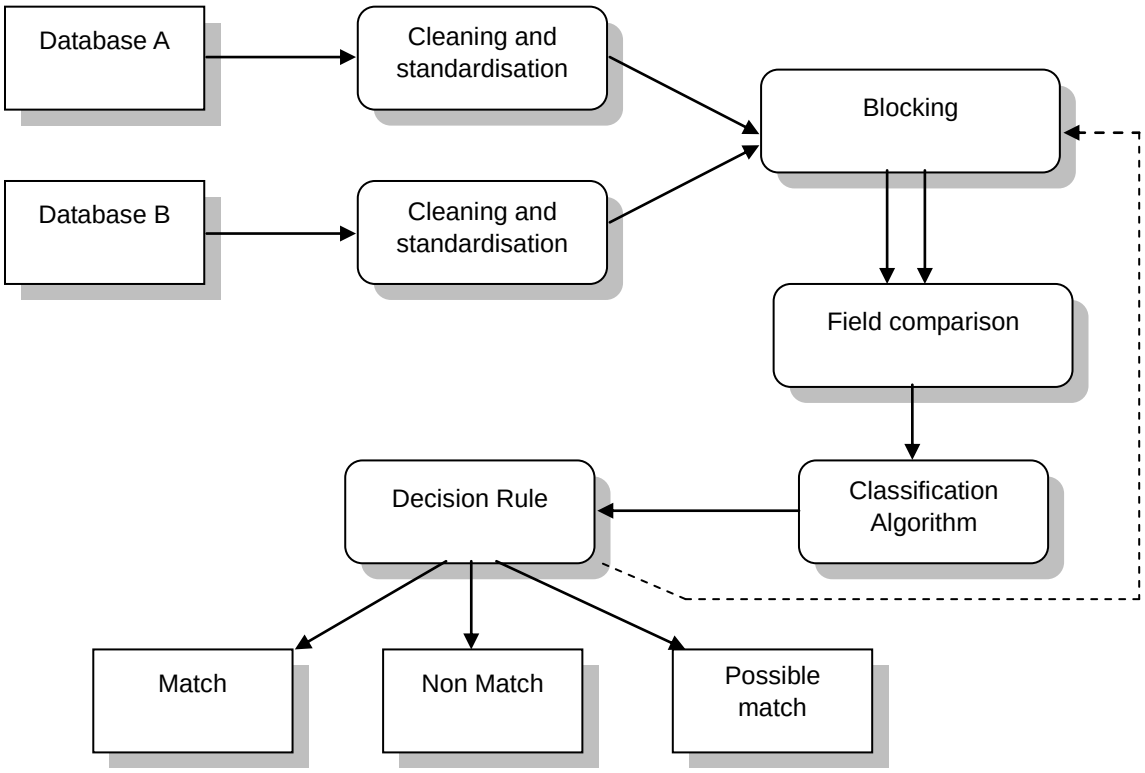
Linking administrative data from different sectors creates a valuable source of information for statistical and research purposes because relationships that previously could not have been considered can be examined. This paper describes the RL actions realized in the Institute of Statistics of Andalusia, the official statistical agency of this region of Spain. The main benefit is the development of a set of tools based on open source software which have been tested on several population databases. This resource will grow to incorporate new functionalities and to merge databases from other sources.

Figure 1 illustrates the Record Linkage process. The cleaning step removes unwanted characters and words, expand abbreviations, or correct misspellings. We have used search tables and correcting lists for performing this first process. The standardization phase refers to methods for breaking free-form fields such as names or addresses into components that can be more easily compared. It can also include methods for putting dates into a standardized format. We have trained Hidden Markov Models (Rabiner, 2007) to automatically segment address and names. Usually, the files are too big to consider every pair in the cross product space of all pairs from two files. It is needed a reduction in the number of pairs through a process called blocking. Newcombe (1988) showed how to reduce the number of pairs by only considering pairs that agreed on a characteristic such as surname or date birth. Our works used a blocking scheme according to the

name variable.

After the data preparation and blocking, the next step computes a set of comparisons between the fields on each pair of records. We have used six fields in our work: first surname, second surname, identification card number, birth date, province code and municipality code. A normalised similarity measure between 1.0 (the strings are the same) and 0.0 (strings are totally different) is usually calculated for each field. There exists a great number of similarity measures (Christen, 2006). We have considered four comparison functions: Exact string, Edit, Jaro and Winkler. Exact measure is 1 when both strings totally agree and 0 when they are not exactly equal. Edit or Levenshtein distance is defined to be the smallest number of edit operations (insertions, deletions and substitutions) required to change one string into another. Jaro measure is based on the number of insertions, deletions and transpositions. Winkler algorithm improves upon the Jaro measure by applying ideas based on empirical studies which found that fewer errors typically occur at the beginning of names.

Figure 1.
General Record Linkage Process.



Febrl (Freely Extensive Biomedical Record Linkage) is the open source platform we have used in our project. It is available under an open source software license (http://sourceforge.net/projects/febrl/), and it contains many recently developed advanced techniques for data cleaning and standardisation, blocking, field comparison, and record pair classification, and offers them into a graphical user interface (Christen, 2008).

Febrl is implemented in Python, a free, object-oriented programming language that is available on all major computing platforms and operating systems. Many organizations use Python, and due to its clear structure and syntax it is also used by various universities for undergraduate teaching in introductory programming courses. Python provides data structures such as sets, lists and dictionaries (associative arrays) that allow efficient handling of very large data sets, and includes many modules offering a large variety of functionalities. Its large number of extension modules facilitates database access and graphical user interface (GUI) development.

Febrl is suitable for the rapid development, implementation, and testing of new and improved record linkage algorithms and techniques, as well as for both new and experienced users to learn about. We have also used R system (R Development Core Team, 2010) to fit some classification algorithms.

Next section describes the classification rules we have used in the empirical evaluation presented in section 3.

2. Record Pair Classification.

The previous steps provide a dataset where each line contains the values of the similarity measures, computed on one pair of records. This dataset must be used to derive a classification algorithm which assigns to record pairs the status match or non-match. Some rules allow a third and intermediate state, possible match. When the training data contains the true status for each pair, a supervised classification can be fitted, otherwise a non-supervised algorithm must be constructed. We have used both types of algorithms.

2.1. Non-supervised rules.

Three non-supervised rules have been considered: Fellegi and Sunter, Farthest First and EM algorithm. This last method has been implemented in R, while the other two modes are available inside Febrl.

The classical Fellegi and Sunter classifier (Fellegi and Sunter, 1969), computes for each pair the sum of similarities into one matching weight, and then it uses two thresholds to classify a record pair into one of the three possible classes: links, non-links or possible links. The two thresholds have manually to be selected by the user, and record pairs that have a matched weight below the lower threshold will be classified as non-matches, record pairs with weights above the upper threshold as matches, and record pairs with weights between these two thresholds as possible matches.

Farthest first clustering defines two clusters (one for matches and one for non-matches) but the parameter "Fuzzy region threshold" also allows the possible match decision. It differs from k-means clustering, also available in Febrl, in the process defining the final centroids. The parameter "Centroid initialization" has different values, being „Traditional" the procedure we have used: first, a random weight vector is chosen as first centroid, and the weight vector furthest away from this centroid is selected as second centroid. Then, the weight vector furthest away from the second centroid is considered a first centroid. This process is repeated 10 times, and the final two centroids selected (assumed to be the weight vectors furthest away from each other) will be the centroids used as in the clustering. This approach has shown to achieve good results in an experimental evaluation (Goiser and Christen, 2006).

EM algorithm applies the classical Expectation-Maximization algorithm to estimate the following probabilities, where γ is a vector of comparisons: $m(\gamma)=P\{\text{Match}/\ \gamma\}$ and $u(\gamma)=P\{\text{Non-Match}/\ \gamma\}$. The values of the ratio $W(\gamma)=m(\gamma)/\ u(\gamma)$ guide the user in the selection on two threshold values, as in Fellegi and Sunter classifier. We have implemented in R this rule for the exact string comparison.

2.2. Supervised rules.

A classification tree (CT) is a set of logical "if-then" conditions which drive each case to a final decision. These conditions can be easily plotted helping us to understand the model. A binary CT is grown by binary recursive partitioning using the response in the specified formula and choosing splits from the set of predictor variables. The split which maximizes the reduction in impurity (a measure of diversity for the outcome in a specific set of nodes) is chosen, the data set is split and the process is repeated. Splitting continues until the terminal nodes are too small to be split. The classification for a vector is computed by a majority class vote in its terminal node. We have been used the rpart package of R (Therneau and Atkinson, 2010), which implements the CART methodology as proposed by Breiman (Breiman et al., 1984). The Gini index (default impurity measure) has been considered as the splitting criterion. The user must tune a fundamental parameter: the number of terminal nodes, called the size of the tree. We have used the 1-ES rule to select the size of the tree.

Random forests (RF) was proposed by Breiman (2001) as a way to combine many different trees. A number of trees are constructed. Each one is grown over a bootstrap sample of the training data set, and a random selection of variables is considered to choose splits in each node. As in bagging, the trees are combined by majority voting, and out-of-bag estimates can also be computed. One important feature of this ensemble method is the availability of some measures to assess the importance of each variable and to identify outlier observations. We have used the R package randomForest (Liaw and Wiener, 2002), which builds 500 trees by default.

2.3. Two-step rule.

The basic idea is to automatically select in a first step weight vectors that very likely correspond to record pairs that are true matches, and weight vectors that very likely correspond to record pairs that are true non-matches, and then use these selected weight vectors for training of a supervised classifier in a second step. Febrl lets to select a number of weight vectors sufficiently close to the vector all of ones, and number of weight vectors sufficiently close to the vector all of zeros. These numbers are controlled by the parameter "Match method". This can either be nearest based, in which case the number of weight vectors nearest to the zero or one vector only has to be given; or it can be set threshold based nearest selection, in which case a numerical threshold has to be given as well.

In the second step, any supervised classification rule may be used. Febrl offers the Support Vector Machine classifier (SVM) and therefore it has been used in our work with radial basis kernel.

3. An empirical comparison.

We present in this section an empirical comparison. It was performed using data from two sources: Vital Statistics and Municipal Register on Inhabitants. Vital Statistics (VS) offer information on births, marriages and deaths that occur in Andalusia during the reference year, beginning the detailed series of tabulations in 1996. The objective is to offer exhaustive information on the above mentioned vital events. The Municipal Register on Inhabitants (MRI) is an administrative register which contains the neighbours of each municipality. It provides a population count and information on demographic structure. A continuous and computerized census management based on National Institute of Statistics coordination has recently been implemented.

10000 records of VS and 8068 of MRI were randomly selected for our empirical comparison. We have used six fields in our work: first surname, second surname, identification card number, birth date, province code and municipality code. After the cleaning and standardization phase of both files, a blocking scheme based on the name value was performed, reducing to 1757210 the number of record pairs. This set was randomly split into training (75%) and test sets (25%) for obtaining reliable performance measures. The analysis of results was based on the following measures:

Accuracy is the rate of correct decision; precision is the rate of link decisions inside the set of actual links; recall is the true link rate, also known as sensitivity; F is the harmonic mean of precision and recall; specificity is the true non link rate. Tables 1 to 5 contain the test results for the six classification rules. Table 6 displays the measure of importance for each variable in the Random Forests models.

Classification trees, Random Forests and EM with exact string provide the best results. It is remarkable that the traditional EM approach remains as a powerful classification rule against the modern machine learning models tried in our study. From table 6 we can see that identification card and birth date are the main variables in the Random Forests model.

The analysis of the results also suggests that the best results are obtained with exact string, and it also exhibits lower variability.

Table 1.

Fellegi Sunter

Measure	Exact	Edit	Jaro	Winkler
Accuracy	0.999	0.999	0.999	0.999
Precision	0.969	0.959	0.957	0.893
Recall	0.883	0.911	0.911	0.912
F	0.924	0.935	0.933	0.902
Specificity	0.999	0.999	0.999	0.999

Table 2.

Farthest first

Measure	Exact	Edit	Jaro	Winkler
Accuracy	0.999	0.995	0.985	0.918
Precision	0.969	0.473	0.220	0.045
Recall	0.891	0.923	0.922	0.921
F	0.928	0.625	0.356	0.087
Specificity	0.999	0.995	0.986	0.918

Table 3.

SVM

Measure	Exact	Edit	Jaro	Winkler
Accuracy	0.999	0.986	0.976	0.969
Precision	0.961	0.227	0.146	0.115
Recall	0.920	0.925	0.924	0.924
F	0.940	0.364	0.252	0.205
Specificity	0.999	0.986	0.976	0.969

Table 4

Classification trees

Measure	Exact	Edit	Jaro	Winkler
Accuracy	0.999	0.999	0.999	0.999
Precision	0.999	0.999	0.999	0.999
Recall	0.999	0.999	0.999	0.999
F	0.999	0.999	0.999	0.999
Specificity	0.997	0.989	0.983	0.995

Table 5.
Random Forests and EM algorithm

Measure	RF Exact	RF Edit	RF Jaro	RF Winkler	EM Exact
Accuracy	0.999	0.999	0.999	0.999	0.999
Precision	0.999	0.999	0.999	0.999	0.999
Recall	0.999	0.999	0.999	0.999	0.999
F	0.999	0.999	0.999	0.999	0.999
Specificity	0.994	0.997	0.996	0.996	0.997

Table 6.
Random Forests: Variable importance

Variable	Exact	Edit	Jaro	Winkler
First surname	1516.7	1307.2	1519.1	1481.5
Second surname	1532.2	1548.4	1641.7	1529.6
Birth date	3428.2	4118.9	3727.7	4102.6
Identification card	5559.6	5631.2	5583.8	5436.9
Province code	167.2	115.8	119.2	150.5
Municipality code	780.4	527.2	603.1	583.2

REFERENCES

Breiman, L. 2001. Random Forests. Machine Learning 45(1), 5-32.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. 1984. Classification and Regression Trees. Wadsworth and Brooks: Belmont.

Christen, P. (2006). A comparison of personal name matching: Techniques and practical issues. In: Workshop on Mining Complex Data (MCD), IEEE ICDM'06, Hong Kong.

Christen, P. (2008). Febrl – A Freely Available Record Linkage System with a Graphical user Interface. In: HDKM '08 Proceedings of the second Australasian workshop on Health data and knowledge management. Volume 80.

Fellegi, I. P., and Sunter, A. B. (1969), "A Theory for Record Linkage," Journal of the American Statistical Association, 64, 1183-1210.

Goiser, K. and Christen, P. (2006). Towards automated record linkage. In Australasian Data Mining Conference (AusDM'06), Conferences in Research and Practice in Information Technology (CRPIT), volume 61, pages 23–31, Sydney.

Handbook of Vital Statistics Systems and Methods, Volume 1: Legal, Organisational and Technical Aspects, United Nations Studies in Methods, Glossary, Series F, No. 35, United Nations, New York 1991.

Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. R News 2(3), 18-22.

Newcombe, H. B. (1988). Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business, Oxford: Oxford University Press.

R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.r-project.org.

Rabiner, L.R. (2007). A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77 (2), 257-286.

Therneau, T.M. and Atkinson, B. R port by Brian Ripley. (2010). rpart: Recursive Partitioning. R package version 3.1-46. http://CRAN.R-project.org/package=rpart