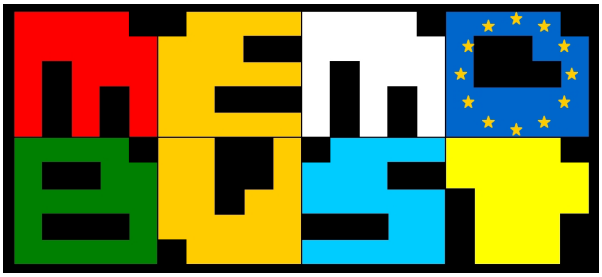

This module is part of the

Memobust Handbook

on Methodology of Modern Business Statistics

26 March 2014

Theme: Object Matching (Record Linkage)

Contents

General section

1. Summary

The aim of object matching (more commonly known as record linkage or as record matching) is to match the same units that are represented by records in two different files. This is to be contrasted with synthetic (or statistical) matching where the aim is to match similar, but usually different, units. Depending on the kind and quality of the information available a suitable matching method should be identified. In case object identifiers of good quality are available in both files, it is quite straightforward to use these to find the records matching on this key. Complications may arise when such object identifiers are not present. In that case one should investigate if object characteristics are present in both files that can be used for finding matches. Several methods exist that deal with this situation. The aim of the present module is to provide a context and overview of the various matching methods, and to give pointers to the specialised method modules in this handbook dealing with these methods.

2. General description

2.1 Purpose of matching

The increasing demand for timely, detailed and high-quality statistics combined with the obligation to use existing registries as much as possible makes it necessary to find alternative ways to produce statistics, such as by matching information from different files. Registries, for example, are not designed to produce statistics. To produce the desired statistics anyway, it is necessary to match registries and survey data to create more usable data sets. In this context, longitudinal data must also be taken into account. On the output side, there is more of a need to present events in their mutual relationships and not only as separate statistics. Matching of files makes it possible to publish over broader themes and to develop new output.

Data matching contributes, for example, to the following:

- Faster publishing of new output;
- Better quality of data through, for example, mutual confrontation;
- Reduction of the survey pressure and therefore lower costs for the respondents;
- Reduction of the costs of the NSI because it no longer needs to conduct surveys in a particular areas.

Data matching therefore supports the main goals of the NSI, such as creating new output, generate less survey burden, make better use of administrative sources and operate more efficiently.

Recent information on matching can be found among in Herzog et al. (2007) and the documents of the ESSnet project on Integration:

http://www.cros-portal.eu/content/data-integration-1

and

http://www.cros-portal.eu/content/work-packages-and-executive-summary.

Willenborg and Heerschap (2012) was used as a source of the present module (as well as of several of the modules on matching in this handbook).

2.2 What is matching?

Matching is about combining information from two or more records (each representing units in a target population), which are believed to relate to the same unit (or object), such as a person, business or region (see Newcombe, 1988). Normally in the matching process, two similar records, present in two different files (known as matching files) are combined, based on various criteria and preconditions. It should be stressed that this type of matching is different from that in statistical matching, where the aim is to match objects that are similar but not identical. Statistical matching therefore, although in execution being very close to the type of matching considered here, is more akin to imputation. (See the theme module "Micro-Fusion – Statistical Matching" in the handbook.)

The most direct case of matching concerns object identity matching. Here one attempts to join objects represented in different data files using identifiers for the objects. For this purpose, a matching key is used consisting of several (key) variables that both files have in common. The matching criterion can then be: 'exactly the same scores on the matching key'. This is a relatively simple (but important) situation that often exists in practice.

Object identifiers suitable for object identifier matching are not always available in matching situations. However, it may be the case that object characteristics are present in the files to be matched, that allow certain objects (records) to be matched. As these characteristics are not key values that identify objects uniquely, it is possible that for a given object there are several candidates. In this case the matching takes place in two steps:

1. It is determined which records are *matching candidates*, potential matches, so to speak, and
2. From all possible matching candidates, the *best subset* is selected, which satisfies certain criteria (preconditions), for example, that no single record is matched with two or more records.

It is possible to simply indicate which objects are matching candidates or not, or it may be possible to differentiate in the strength of being matching candidates, using matching weights to express the strength of the matching. Matching candidates that have more characteristics in common are than stronger matches than those with less. These matching weights may also be probabilities, derived from a probabilistic matching model.

The decision to match or not to match objects (thus determining which matching candidates are considered matches) is generally made by a matching programme. If the matching takes place interactively or manually, a matching specialist takes these decisions.

In Figure 1 a schematic view of the matching process is presented. It indicates that in case two files are matched, first matching criteria have to be identified, including the choice of matching variables, when two objects are considered matches or not (in case object identifiers are used) or how to calculate the strengths of possible matches (matching candidates), expressed in matching weights. The matching that is then carried out yields matching candidates. From these the final matches are determined. It generally yields three subgroups of the group of matching candidates: those matching candidates that are considered as matches, those matching candidates that are considered as non-matches, and those

matching candidates for which it is not so clear whether they match or not (the doubtful cases). The first group is the one that will be used for further analysis. The second group is not. In case the third group is big, it may be that the matching is repeated, this time with (slightly) different matching criteria, in the hope that the yield (the size of first group, the matches) may be higher.

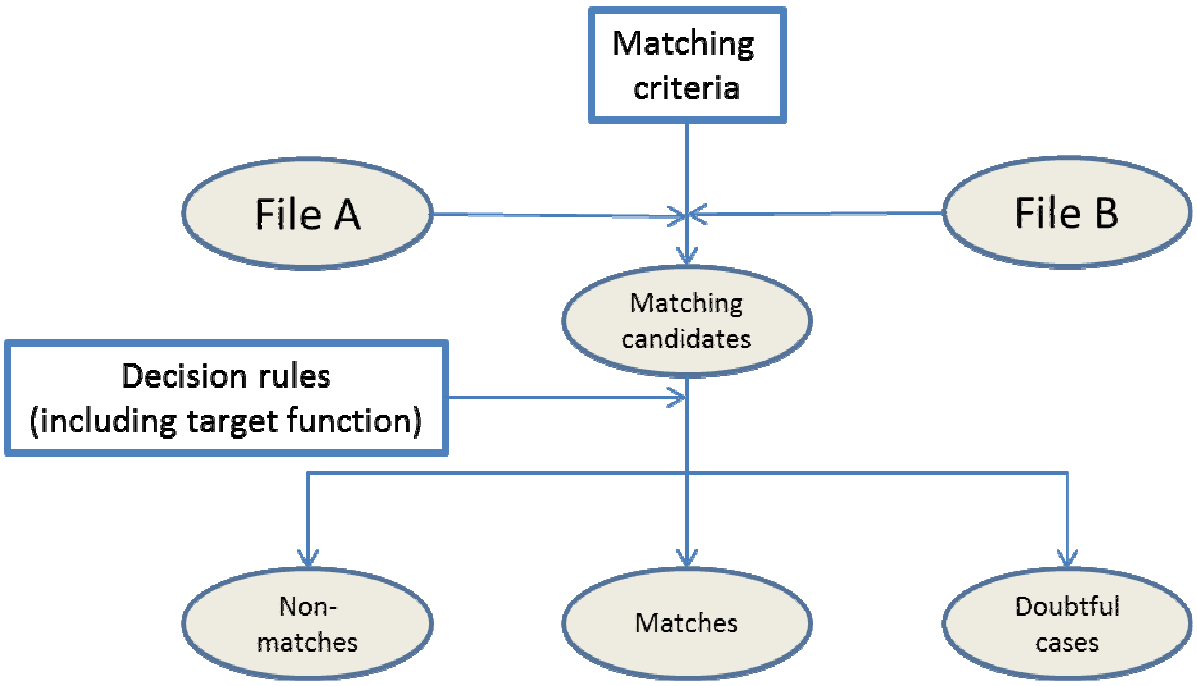

Figure 1. Main ingredients in matching.

In the next section we consider various matching methods in a bit more detail. But its main objective is to refer to the various modules in the handbook that deal with these methods in more detail.

2.3 Overview of object matching

In the handbook several matching methods are discussed focussing on matching identical objects.

The first one is uses object identifiers for matching. In this case for the objects to be matched object identifiers (also known as keys) are available. They have the property that they uniquely identify objects. They are ideal for matching objects, provided they are free of error. This is a matching method that is typical for, but not limited to, databases, where it is known as 'joining'. This method is important as it is used frequently in practice. It is the simplest of the matching methods that we address in this report. For more information see the method module "Micro-Fusion – Object Identifier Matching".

In practice object identifiers are not always available. But characteristics of objects may be available for matching. That brings us to the next form of object matching, namely that which uses object characteristics of objects. In fact, there is no single method for this kind of matching. We distinguish between two types of methods. The one uses no matching weights and the other does to distinguish in the strength of potential matches.

The first group of methods of methods dealing with object characteristics does not use matching weights to differentiate between the strength of matches. It is elaborated in the method module "Micro-Fusion – Unweighted Matching of Object Characteristics". The second group of methods dealing with object characteristics is uses matching weights to express differences in strengths of potential matches The matching weights use to express the strength of potential matches can be calculated in various ways, depending on the problem at hand. One can use a metric (or distance function) or measure of dissimilarity to quantify how object characteristics differ. This class of matching methods is elaborated in the method module "Micro-Fusion – Weighted Matching of Object Characteristics".

A special case of weighted matching is probabilistic record linkage. In this case the matching weights are derived from a probabilistic matching model. More details on this type of matching can be found in this handbook in the theme module "Micro-Fusion – Probabilistic Record Linkage".

A special case of probabilistic matching that deserves special attention is a classical method proposed by Fellegi and Sunter (1969) and refined by Jaro (1989). In the handbook it is discussed in the method module "Micro-Fusion – Fellegi-Sunter and Jaro Approach to Record Linkage".

2.4 Matching errors

The matching of two files may lead to errors for various reasons (see also Section 2.5). After the matching candidates have been identified and the matches selected from them, two kinds of errors may result:

- Mismatches: records that are matched, but are not actually associated with the same objects.
- Missed matches: records that are not matched, but that are actually associated with the same objects.

Table 1 contains an overview of the various matching errors including the various names that are being in the literature used to indicate them.

Table 1. Object matching errors

	Objects associated with same unit	**Objects associated with different units**
Objects matched	- good result - rightly matched - correct match	- mismatch - false positive match - type I error - erroneously matched
Objects not matched	- missed match - false negative match - type II error - erroneously not matched	- good result - rightly not matched - correct unmatch

In practice, it is usually unknown whether a match of two records is correct, or a mismatch, or when two records should have been matched because they pertain to the same object (missed match). Nevertheless, it is useful to distinguish these errors.

2.5 Why is matching complex?

At first glance, the matching of files seems to be a simple task. In practice, however, this is seldom the case, especially in the context of business statistics. The following causes contribute to this circumstance:

- The *quality and the structure of the data* in the files to be matched. It will seldom be the case that the data provided, and therefore also matching variable data, do not contain 'noise'. During processing, for example, observation and processing errors, such as typing errors, can occur. Consequently, it is possible that records that actually do correspond do not match, or vice versa. With respect to the structure of the data provided, it is possible, for example, for the scores of the matching variables to be good in both records, while they are represented in such a way that it is difficult to compare these with one other via automation. All of these aspects make the pre-processing stage important. This is where both the quality and the structure of the data can be adapted and improved, insofar as is necessary for matching.
- The *units of files to be matched may differ*, but still can be derived from one another. Consider, for example, a file with Business Units that must be linked with a file with Enterprise Groups. In this context, a matching table should be used that sets out the relationship between both units.
- The use of *different domains or classification divisions* for the matching variables. Here as well, it is desirable for the matching process that the domains or classifications are compatible.
- The *time dimension*. The matching variables or units are dynamic and were observed at different moments in time. This could be the case, for example, for businesses. In the time between two different observations, which are saved in the two different files, the enterprise may have split or merged, while it still has the same identifier or matching variable. In the matching process, this would seem to refer to the same enterprise, while in reality, the enterprise may not be the same anymore.

2.6 Matching applications

Examples of matching applications in the statistical process are the following:

- ***Micro-fusion***. In this process, different pieces of data are confronted with each other, and a variety of differences about businesses may become apparent. The aim is then to explain and eliminate these differences. Confronting the data is only possible after the files have been matched. See the various modules in the handbook on micro-fusion, in particular those dealing with differences in the data and how to reconcile them.
- ***Input matching***. Starting with the building of a statistical frame. Usually, a combination of sources is needed to compile such a frame or 'backbone', for example, the General Business Register. In the Netherlands, for example, matched data from the Chamber of Commerce and Tax

Administration are used. For more information on this see the modules of the topic "Statistical Registers and Frames" in the handbook.

- ***Statistical matching***. Statistical (or synthetic) matching is concerned with filling in missing values in a file, and an auxiliary file is used for this purpose. Information from *similar* objects is used to fill in the missing values. So the goal of statistical matching is to match similar objects, not (necessarily) identical ones. The method can be viewed as an imputation method. See the theme module "Micro-Fusion – Statistical Matching".
- ***Allocation of CATI interviewers to sample elements.*** The matching is carried out for the purpose of interviewing businesses, say. Here, the problem is deciding which interviewer should call which business at what time. The matching between interviewer and business to be called is done in several steps. First the deployment of the interviewers is scheduled. When they are at work they get telephone numbers of businesses assigned that they should call for CATI interviews. For more information on this see the theme module "Data Collection – CATI Allocation".
- ***Coding.*** In this process, descriptions given by respondents in their own words are matched with codes from a classification. One of the problems here involves matching of words, while knowing that the respondents could have potentially made spelling or grammatical errors or used synonyms, hyponyms or hypernyms. See the modules of the topic "Coding" in this handbook for more information on this subject.

3. Design issues

4. Available software tools

Data matching is virtually impossible without the use of a specialised software package. Some examples of matching software tools are the following:

- **Trillium** (Harte-Hanks; www.Trilliumsoftware.com).
- **SSA NAME3** (Search Software America; www.searchsoftware.com).
- **IQ-Matcher** (Intech Solutions; http://www.intechsolutions.com.au).
- **Other matching tools** include: GDriver (US Census Bureau/Winkler), Relais (Istat), LinkageWiz, Tailor (a record linkage toolbox), NameSearch from Intelligent Search Technology, PA Oyster Engine, Fril, OxLink and Alta.

5. Decision tree of methods

Figure 2 presents a decision tree for the application of the various matching methods considered in the handbook.

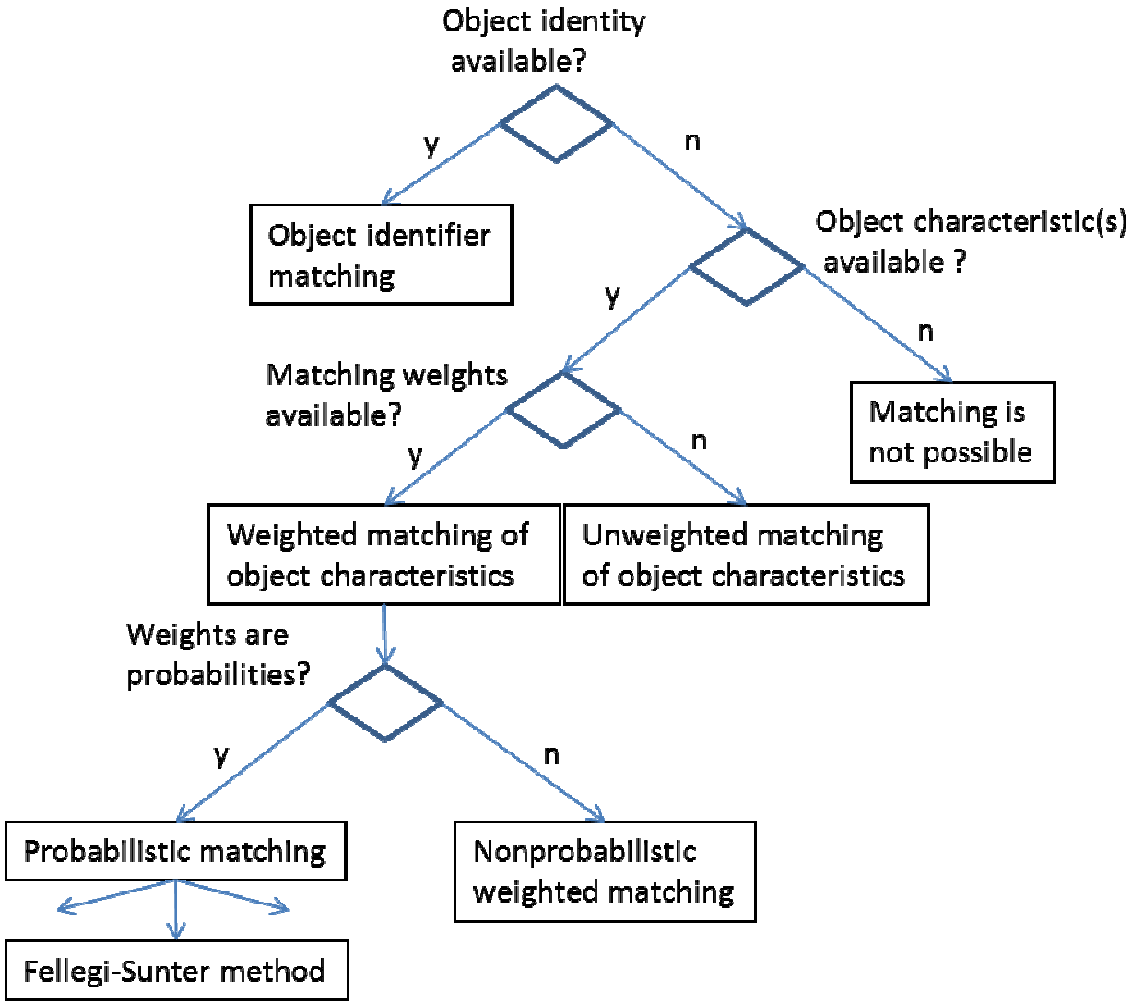

Figure 2. Overview of different matching methods.

6. Glossary

For definitions of terms used in this module, please refer to the separate "Glossary" provided as part of the handbook.

7. References

Fellegi, I. P. and Sunter, A. B. (1969), A theory for record linkage. *Journal of the American Statistical Association* **64**, 1183–1200.

Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007), *Data quality and record linkage techniques*. Springer.

Jaro, M. A. (1989), Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association* **84**, 414–420.

Newcombe, H. B. (1988), *Handbook of record linkage*. Oxford University Press.

Willenborg, L. and Heerschap, N. (2012), *Matching*. Contribution to the Methods Series, Statistics Netherlands, The Hague.

Interconnections with other modules

8. Related themes described in other modules

1. Statistical Registers and Frames – Main Module
2. Data Collection – CATI Allocation
3. Micro-Fusion – Data Fusion at Micro Level
4. Micro-Fusion – Probabilistic Record Linkage
5. Micro-Fusion – Statistical Matching
6. Coding – Main Module
7. Imputation – Main Module
8. Dissemination – Dissemination of Business Statistics

9. Methods explicitly referred to in this module

1. Micro-Fusion – Object Identifier Matching
2. Micro-Fusion – Unweighted Matching of Object Characteristics
3. Micro-Fusion – Weighted Matching of Object Characteristics
4. Micro-Fusion – Fellegi-Sunter and Jaro Approach to Record Linkage
5. Micro-Fusion – Statistical Matching Methods

10. Mathematical techniques explicitly referred to in this module

1.

11. GSBPM phases explicitly referred to in this module

1. Phase 5.1 Micro-integration.
2. Phase 5.2 Coding.

12. Tools explicitly referred to in this module

1. Alta.
2. Fril.
3. GDriver.
4. IQ-Matcher.
5. Linkage Wiz.
6. NameSearch.
7. Oxlink.

8. PA Oyster Engine.
9. Relais.
10. SSA Name3.
11. Tailor.
12. Trillium.

13. Process steps explicitly referred to in this module

1. Integration / micro-aggregation of information
2. Coding
3. Allocation of sample units to interviewers
4. Dissemination of information
5. Statistical (synthetic) matching

Administrative section

14. Module code

Micro-Fusion-T-Object Matching

15. Version history

Version	Date	Description of changes	Author	Institute
0.1	30-06-2012	first version	Leon Willenborg, Rob van de Laar	CBS (Netherlands)
0.2	02-07-2012	second version	Leon Willenborg, Rob van de Laar	CBS (Netherlands)
0.3	11-07-2013	third version	Leon Willenborg	CBS (Netherlands)
0.4	09-08-2013	new version (using review comments)	Leon Willenborg	CBS (Netherlands)
0.4.1	21-08-2013	minor revisions	Leon Willenborg	CBS (Netherlands)
0.5	03-11-2013	new version (using EB review comments)	Leon Willenborg	CBS (Netherlands)
0.5.1	18-11-2013	preliminary release		
1.0	26-03-2014	final version within the Memobust project		

16. Template version and print date

Template version used	1.0 p 4 d.d. 22-11-2012
Print date	21-3-2014 17:56