Method Series

Theme: Matching

12

Leon Willenborg and Hico Heerschap

Statistics Methods (201203)

Statistics Netherlands

The Hague/Heerlen, 2012

Explanation of symbols

.	data not available
*	provisional figure
**	revised provisional figure (but not definite)
x	publication prohibited (confidential figure)
–	nil
–	(between two figures) inclusive
0 (0.0)	less than half of unit concerned
empty cell	not applicable
2011–2012	2011 to 2012 inclusive
2011/2012	average for 2011 up to and including 2012
2011/'12	crop year, financial year, school year etc. beginning in 2011 and ending in 2012
2009/'10–2011/'12	crop year, financial year, etc. 2009/'10 to 2011/'12 inclusive

Due to rounding, some totals may not correspond with the sum of the separate figures.

Publisher
Statistics Netherlands
Henri Faasdreef 312
2492 JP The Hague

Prepress
Statistics Netherlands
Grafimedia

Cover
Teldesign, Rotterdam

Information
Telephone +31 88 570 70 70
Telefax +31 70 337 59 94
Via contact form:
www.cbs.nl/information

Where to order
E-mail: verkoop@cbs.nl
Telefax +31 45 570 62 68

Internet
www.cbs.nl

ISSN: 1876-0333

60165201200 X-37

Table of Contents

1. Introduction to the theme 5

1.1 Introduction and background 5
1.2 Place in the statistical process 6
1.3 Scope and relationship with other themes 6
1.4 Deviation from the standard breakdown of methods 7
1.5 Reading guide 7
1.6 Concepts and definitions 8

2. Overview of the matching problem 13

2.1 What is matching? 13
2.2 What makes matching so complex? 15
2.3 Steps in the matching process 16

3. Graphs and metrics for matching 21

3.1 Graphs 21
3.2 Metrics 24

4. Matching theory 26

4.1 Introduction 26
4.2 Choosing between the matching methods 29
4.3 Matching models based on graphs 30
4.4 Matching problems in graphs 31
4.5 Working methods 33

5. Matching based on a primary key 37

5.1 Short description 37
5.2 Applicability 37
5.3 Detailed description 38
5.4 Examples 38
5.5 Quality indicators 39
5.6 Variation 39

6. Matching based on secondary keys, without matching weights 40

6.1 Short description 40
6.2 Applicability 40
6.3 Detailed description 41
6.4 Example 43
6.5 Quality indicators 44
6.6 Variant: use of a distance function 44

7. Matching based on secondary keys, with matching weights 47

7.1 Short description 47
7.2 Applicability 47
7.3 Detailed description 48

7.4 Example 53
7.5 Quality indicators 54
7.6 Variants 54
8. Matching software and IT considerations 55
8.1 Matching software 55
8.2 IT considerations 58
9. Special subjects 59
9.1 Large files 59
9.2 Determining matching parameters 59
9.3 Matching related units 60
9.4 Dealing with ‘remainders’ 61
9.5 Matching personal details 61
9.6 Matching business data 63
10. References 66
Appendix A. The Fellegi-Sunter model 68
Appendix B. More about metrics 70
Appendix C. Considerations when selecting matching software 72

1. Introduction to the theme

1.1 Introduction and background

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. Examples of this are: the themes on ageing and globalisation. As such, we are able to better satisfy the current needs of the users of these statistics.

Data matching contributes, for example, to the following:

- The faster publishing of new output;
- A better quality of data, through, for example, mutual confrontation;
- The reduction of the survey pressure and therefore lower costs for the respondents;
- The reduction of the costs of Statistics Netherlands because it no longer needs to conduct surveysI in a particular areas.

Data matching therefore supports the main goals of the Statistics Netherlands, such as new output, less survey burden, better use of administrative sources and lower costs. The extent to which data matching contributes to more efficiency, in the sense of fewer FTEs, is difficult to determine. On the one hand, this will lead to savings if, for example, the number of the organisation's own surveys are limited further. On the other hand, the matching of files and the analysis of the results will demand extra capacity. It is clear that other competencies will be needed for this activity. Acquiring knowledge concerning the files to be matched is also important and requires significant efforts.

This report describes the *matching methodology*.[1] This concerns mainly the problem of bringing together information from records originating from two or more files that relate to the same units, observed at virtually the same time. This can be a relatively simple task, especially if there is a common and unambiguous matching key for the units in both files, for which the scores of the matching key are also reliable. In terms of people, you can view the citizen's identification number as a clear matching key. However, it can also be much more difficult, for example, when such a clear matching key does not exist, but when there are several secondary matching variables such as name, address, date of birth and age that are jointly present, but which do not always have to have equally reliable scores. Or the case that the matching key does not have exactly the same variables, but similar ones, such as with a slightly different domain. An even more difficult situation arises if

[1] This document is a translation of a Dutch report that was written for internal use. For that reason certain references are made that apply to CBS specific situations or objects. We did not want to make the translation more general, because that would increase the writing effort. Where necessary we explained CBS-specific terminology or situations in the glossary or in footnotes.

the units in both files are not the same, for example, as a result of dynamics in populations. In addition to the birth and death of units, units can age or transform into other units. Examples of this are when businesses merge or split.

1.2 Place in the statistical process

Data matching is not limited to one specific place in the statistical process. In fact, data matching can be performed at any place in the statistical process. On the input side, it begins with the building of the statistical frame. Usually, a combination of sources is needed to compile such a frame or 'backbone'. That is true, for example, for the General Business Register in economic statistics. In the Netherlands, for example, matched data from the Chamber of Commerce and Tax Administration are used. In the processing stage, data matching can be utilised in different ways; for example, as extra information in checking the quality of the data or when deriving data,in imputation. With regard to the output, this concerns mainly obtaining new information by combining data from different sources.

1.3 Scope and relationship with other themes

This report first discusses the matching methods whose goal is to create a connection between data from the same units, when this data is represented in different files.

Matching is related to other components of the Methods Series, such as:

- ***The integration/micro integration of information***. In this process, different pieces of data are confronted with each other, and a variety of differences become apparant. These differences are explained and then eliminated. Confronting the data is only possible after the files have been matched;
- ***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 words, while knowing that the respondents could have potentially made spelling or grammatical errors or used synonyms, hyponyms or hypernyms.
- ***Allocation of sample units to interviewers.*** The goal here is to match contact information from sample units (people, businesses) with interviewers in order to conduct interviews. For example, CAPI interviews make use of the residential addresses of people drawn into a sample. They must be visited by interviewers so that they can be interviewed. When assigning addresses to interviewers, the wish was to take account of an interviewer's maximum interview capacity and travel times to residential addresses of the sample people. The interview capacity per interviewer must be respected while the travel costs are to be minimised.
- ***Dissemination of information.*** Data matching is necessary to see and present interrelationships in statistical data.

Exclusion:
A method that at first sight seems to be a matching method, and which is known as *statistical or synthetic matching*, is actually an imputation method. The intention behind this method is different from the matching methods discussed in the present report. Statistical matching is concerned with filling in missing values in a file, and an auxiliary file is used for this purpose. Information from

similar units is used to fill in the missing values. So, in this method similar units, not identical ones, are used. For this reason this method is *not* discussed here. For more information on statistical matching, see D'Orazio, Di Zio and Scannu (2006) and De Jong (1991).

1.4 Deviation from the standard breakdown of methods

The discussion of the matching methods in this report deviates somewhat from the standard discussion of this subject in the literature, where a distinction is made between deterministic (or exact[2]) matching and probabilistic matching. Deterministic matching involves all the methods that have an unambiguous rule for when two records match or when they do not. Probabilistic matching uses a probability model on the basis of which a match of two records is either made or not. For this matching method use is often made of a model proposed by Fellegi and Sunter (1969). See Appendix A for a short description of this model.

The problem with this breakdown is that deterministic matching also includes matching variants that are evidently intended to 'take care of' errors in the data. For example, if the matching method involves matching two records if their scores agree on four of the five matching variables. In our approach, this is a matching method that makes use of a metric, which is actually used to take account of errors in the data. Probabilistic matching, however, in our approach, is only one of the models to determine weights for candidate matches.

1.5 Reading guide

In this report, Chapter 2 starts with a general description of what matching is and what makes matching so complex in practice. The success of matching depends not only on the selected method, but also to a significant extent on the preparations made and final processing performed. For this reason, Section 2.3 takes a more extensive look at the different steps in the matching process.

Before the theory of matching and the matching methods themselves are discussed, Chapter 3 provides a short introduction to graphs and metrics, because these play a central role in describing the matching methods in the subsequent chapters.

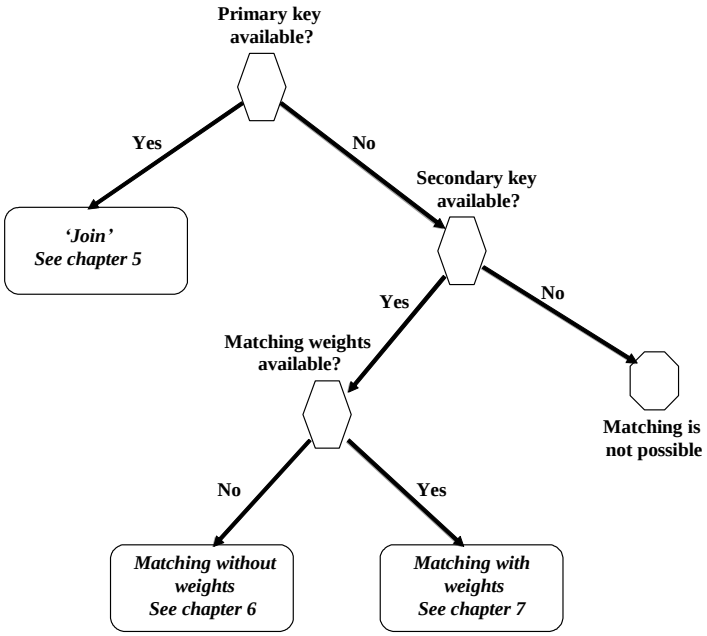
Chapter 4 examines the theoretical aspects of matching in a more general sense. Using the theory, it is possible to clearly demonstrate the similarities and differences of the various matching methods. The theory is also used to take a look at the different matching strategies that exist, and how different optimisation models can be formulated. These models differ with regard to the objectives and preconditions that one can aim for when matching files.

Chapters 5 to 7 discuss three matching methods. First, a matching method is described that is typical for, but not necessarily limited to, databases. This takes place using primary keys and is also called 'joining'. This method is important because it is used frequently in practice. It is the simplest

[2] Because the designation of 'exact' is so misleading in this context, we prefer not to use it in this document. An 'exact match' suggests that it is free of errors. However, that does not have to be the case at all. The matching rule can be exact, but applying this rule to matching files with errors in their matching variables can produce incorrect matches. Moreover, probabilistic matching also utilises an exact rule. In short, the adjective 'exact' tends to create more confusion than clarity.

of the matching methods that we address in this report. Next, we discuss two matching methods that are used in less favourable conditions than joining. These are matching methods based on secondary keys. In these methods, we make a distinction between matching without matching weights (Chapter 6) and matching with matching weights (Chapter 7). We can formally explain matching without matching weights as a special case of matching with weights (for example, with all weights equal to 1). By using matching weights, it becomes possible to separate potential matches from one another in terms of 'matching strength'. Figure 1.1 shows the distinctions between the various matching methods in a diagram.

Figure 1.1 Overview of the different matching methods



In practice, software is generally used for matching, if only because of the size of the files to be matched and the staggering number of matching candidates that arises as a result. Chapter 8 briefly discusses several software packages that can be used to perform matching.

Chapter 9 examines several practical aspects of matching that arise on a regular basis.

The report concludes with a literature list and three appendices. Appendix A describes the matching method of Fellegi and Sunter, which does not actually play an important role in this report, but is very significant historically. Appendix B examines several issues relating to metrics. Appendix C provides a list of questions that are important to consider when selecting the appropriate matching software.

1.6 Concepts and definitions

Table 1.1 sets out the major concepts that play a role in this report, with a short explanation of their meaning. The specific abbreviations used in this report are also listed in this table, along with their meanings.

Table 1.1: Explanation of some concepts and concepts and abbreviations that relate to matching

Concept	Description
Atomic unit	See: Simple unit
BEID	The unique identifier (at Statistics Netherlands) of so-called business units in the General Business Register (ABR). The business unit forms, together with the Enterprise Group and to a lesser extent the legal entity, the statistical framework on which the economic statistics of Statistics Netherlands are compiled.
Bipartite digraph	A digraph $G = (V, \overline{E})$ where V is the set of points and A is the collection of directed edges (arrows), in which the set of points V is composed of two disjoint subsets V_1 , V_2 . Each arrow a in $\overline{E}$ in such a graph has the characteristic that one of the points of an arrow lies in V_1 and the other in V_2 . In this document, we deal with a special subclass of bipartite digraphs, namely those for which all arrows run from V_1 to V_2 .
Bipartite graph	A graph $G = (V, E)$ where V is the set of points and E is the collection of edges, for which the set of points V is composed of two disjoint subsets V_1 , V_2 . Each edge in such a graph has the characteristic that one of the points of e lies in V_1 and the other in V_2 .
Blocking variable	A variable that is used to partition matching files, that is, divide in a number of subfiles, with the intention of reducing the search space. If the blocking variable, for example, is a residential municipality for a matching problem where people are matched, then this means that only people living in the same municipality (at a certain time) will be matched.
BSN	The acronym for *Burger Service Nummer*, in Dutch the citizen's identification number. .
Composite unit	A unit that is composed of units from a lower order. A household is an example of a composite unit; 'persons' are the simple units from which 'households' are composed..
Connected component (of a graph)	A maximal connected subgraph of a graph.
Connected graph	A graph in which all points are connected and therefore form a single component is called a connected graph.
Cut-off value	A value to limit the matching weights (upwards or downwards). As a result, it is possible, for example, to exclude pairs of records that have an overly high matching weight as candidate matches, because they are not sufficiently similar. Or, conversely, by increasing the cut-off value, it is possible to obtain more candidate matches, because records that are less similar are also considered as matching candidates.
DBMS	Database Management System. Examples of DBMSes are Oracle, MySQL, Sequel Server, MS Access, Postgress.
Deduplication	Taking the duplicate records out of a file, one by one, that occur multiple times, and that all relate to the same unit (in a certain period).
Degree	The degree of a point in a graph is the number of edges in the graph connected to this point.
Degree restriction	Limitation with respect to the degree of part of the points of a graph.
Deterministic matching	A matching technique that does not utilise a probability model. This is the, case for joining, which is matching based on primary keys. But if there are errors in the values of these keys, the resulting matches are likely to be incorrect. Applied in the context of matching with secondary keys, this concept is confusing and also usually not applicable. Even if a 'deterministic matching rule' is used, it is highly possible that matching errors will be made because errors and irregularities are present in the data. This matching method is therefore used as counterpart to 'probabilistic matching'. This document avoids the use of this concept because it is confusing and can easily lead to misunderstandings.
Direct identifier	A variable that can be used to identify entities. This includes primary keys, but also variables such as the BSN, name, address, etc., that can be used to directly identify entities, but possibly not uniquely. Some direct identifiers (such as the BSN) are suitable for use as primary keys. Others (such as name, address, etc.) are suitable for use as secondary keys. See also: indirect identifier.

Concept	Description
Dissimilarity measure	A measure to express the differences between two objects or entities. Somewhat similar to a metric. Antonym: similarity measure.
Distance function	See: Metric
DSC	Data Service Center. A sevice at CBS that in principal stores and makes available all the files that are produced during al the steps (interim storage points, ISPs) in the productionproces of statistcs.
ETL	Extract Transform Load. A set of operations to make an external data set suitable for further processing, e.g. at a statistical office. These operations can be geared towards converting data formats, adapting new variables, converting the coding used in the data set to the coding used at the statistical office, etc.
False negative match	See: Missed match
False positive match	See: Mismatch
Feasible matching graph	A subgraph of an MC graph that satisfies the criteria that are established for the matching graph. These criteria relate at least to the maximum degree of the points or a part thereof (degree restrictions). The word 'feasible' is used in the sense of 'feasible solution'.
Fellegi-Sunter method	Matching method described in Fellegi and Sunter (1969). See Appendix A for a short discussion of this method.
Foreign key	A key value that occurs in a record but is not suitable to identify the record itself. A foreign key is therefore located outside the key of a data set. The purpose of a foreign key is to make a match with a record in another data set which, for example, includes additional data based on that key. Example: A record from an enterprise, which is identified by a BEID, also has unique code – as a foreign key – included in the region where the enterprise is active. In another data set, the code of the region is the primary key with additional data about the region, such as the number of residents, the average turnover in the region, the square km of the region, etc. In a record with personal details, uniquely identified by a BSN, consider a reference to the enterprise where someone works. In this context, for example, a code (for example, the BEID) can be used. Another data set, where the BEID is the key, contains data about the enterprise where the person works. A foreign key is often, but not necessarily, a reference to another unit type than to which the record itself relates. Consider, for example, data for an employee with a reference to his/her supervisor. Both are of the type 'person' and both can be designated by a staff ID number.
Hamming distance	Distance between two records on a matching key, measured by counting the number of variables with different scores.
Incidence matrix	0-1 matrix J that indicates for a graph $G = (V, E)$ what the relationship is between edges in E and points in V. Suppose $\mid V \mid = n$, $\mid E \mid = m$ and J is the $m \times n$ matrix where $J(i, j) = 1$ if point j lies on edge i, and $J(i, j) = 0$ otherwise.
Indirect identifier	A variable that can be used to identify at least some entities in a population, but which is not a direct identifier. Examples are: place of residence, profession, age, gender. Indirect identifiers are candidates for secondary keys. Variables that are neither direct nor indirect identifiers express, for example, views, opinions, beliefs, etc. Such variables are not suitable for use as secondary matching keys. The scores for units on such variables are generally not public knowledge, and they can also fluctuate over time.
Integer programming	A special case of linear programming, in which the variables that occur in the optimisation model are integers and not real numbers.
Interim storage point (ISP)	A point in the statistical process at CBS where certain files are well documented, stored in the DSC and made avialble for general use
Joining	A form of matching used for databases and in which, for example, matching is based on matching keys being identical. (equi-join).
Key	See Primary key, Secondary key
Linear programming	Abbreviated as LP. This is the area where solutions are sought for problems with linear target functions that must be optimised under linear constraints. In this context, the variables are real-valued. Important subclasses are formed by problems in which all, or some, variables take on values in a finite set (such as {0,1}) or a denumerable set (such as the integer numbers). In this case we are dealing with an important subclass of LP, namely integer programming (IP).
Matching	The process of bringing together data (represented in records) relating to units and spread over two files, based on common or very similar

Concept	Description
	characteristics in the form of primary or secondary key values. This matching can be simple, especially if common primary keys are present in these files. It can also be more difficult, especially if only secondary keys are present, for which the scores can also contain errors, or when these variables are not completely identical.
Matching candidate digraph	A bipartite digraph that represents the possible matches between records from two files. The asymmetry in the digraph, for example, can be a result of the different times to which the matching files relate. The arrows indicate, for example, a possible development from a unit in one file to a unit in another file. The edges may or may not be assigned matching weights. A matching candidate digraph symbolises part of the constraints that exist for a matching problem. Abbreviated as MC digraph.
Matching candidate graph	A bipartite graph that represents the possible matches between records from two files. The edges may or may not be assigned matching weights. A matching candidate graph symbolises part of the constraints that exist for a matching problem. Abbreviated as MC graph.
Matching graph	Graph that is the result of a match. It is a subgraph of the matching candidate graph, with the same set of vertices but.with less edges.
Matching key	One or multiple key variables that are used in two or more files to be matched, for example, to search for records from one file in records from another file. If the matching key is a primary key variable, matching based on the similarity of the key will produce few problems as such. If, however, a matching key is used that consists of several secondary key variables, then the matching will generally be more difficult due to errors (or other anomalies) in scores for these variables. However, errors can also occur in primary keys.
Matching weight	For a graph $G=(V,E)$ a function $w: E \rightarrow [0,\infty)$ is a weight function, which associates a non-negative value G with each edge of the G. When matching, this weight expresses how well/poorly records match. It depends on the situation whether a higher/lower matching weight means that matching candidates fit together better/worse.
MC digraph	Matching candidate digraph (see the relevant description)
MC graph	Matching candidate graph (see the relevant description)
Metric	A metric d for a set X is defined as function $d: X \times X \rightarrow [0,\infty)$, so a non-negative function, with the following properties: 1. $d(x,y)=0$ if and only if $x=y$, 2. $d(x,y)=d(y,x)$ for all x,y in X (symmetry), and 3. $d(x,z) \leq d(x,y)+d(y,z)$ for all x,y,z in X (triangle inequality). Sometimes, instead of property 3. a stronger attribute applies: 4. $d(x,z) \leq \max\{d(x,y), d(y,z)\}$ A non-negative function d that satisfies 1, 2 and 4. Is called an ultra-metric.
Mismatch	A match that has been made erroneously (false positive match).
Missed match	A match that should have been made but was not (false negative match).
Primary key	In database technology, the primary key is the name for a variable or a combination of variables that satisfy the following requirements: - the value of the variable (or the combination of variables) is unique in the table (or data set) and therefore unambiguously defines the record in which it occurs. - the variable (or the combination of variables) is filled in everywhere and therefore cannot be empty. The combination of variables is minimal: by eliminating one of the variables, the record is no longer unambiguously defined. If related tables refer to the table in which the variable (or combination) of variables occur, this is used to establish a relationship between tables. Examples are the BSN and the RIN number for people, and the BEID for businesses. In statistical confidentiality, such variables are also called direct identifiers. Unfortunately, statistical security also refers to variables such as name, address, place of residence, etc. as direct identifiers.Such variables are called secondary keys in this document, however.
Probabilistic matching	Matching of the same units on the basis of scores for the matching variables that do not necessarily have to be the same. The differences of scores can have various causes: 1. There are observational or processing errors in the scores

Concept	Description
	2. The units in the two files were observed at different times, or 3. Matching variables in the different files are not defined exactly the same and possibly have other domains.
Record linkage	Another name for 'matching'; see the relevant description.
Referential integrity	In a relational database, this is the basic principle that is required for internal consistency of the different tables in that database. This means that a table always has a key if it is referenced by another table in a key field, possibly a foreign key field. Database systems guarantee consistency and ensure that a transaction that violates the consistency cannot be performed. Example: there is a table (1) with regional data, identified by the postal code. In another table (2), the postal code is used to indicate the region in which someone lives. Referential integrity ensures that the postal codes in table 2 can always be found in table 1. Furthermore, the postal codes in table 1 may not be eliminated if these occur in table 2, either as a primary, secondary or foreign key.
Remainder	Records that cannot be matched when matching is performed on two files. In some cases, no remainder is desired, and it must be 'eliminated' by making extra matches.
RIN	Record Identification Number. A primary key used by Statistics Netherlands to replace keys also known outside Statistics Netherlands (such as the BSN). The reason to use a RIN is based on privacy considerations. It is then impossible to match the file in which they are used (and from which the original keys have been removed) for matching with external files.
Secondary key	A combination of variables that can be used in the identification of units, but which are not used as a primary key. Often, this concerns variables (or a combination thereof) such as name, address, place of residence, date of birth, profession, education, gender, etc. None of these variables can identify the record by themselves, but the combination can be used as a proxy for a primary key, if this is missing. In statistical security, such variables are also called identifiers or indirect identifiers.
Similarity measure	A measure that indicates the extent to which two units are similar. This type of measure (or its complement: the dissimilarity measure) is also used in the multivariate analysis, for example, for clustering. See also dissimilarity measure.
SLA	An agreement with clear appointments between supplier and user of a service or product,
Simple unit	A unit that (for the matching problem in question) is not composed of units of a lower order, also called simple (or atomic) unit. For Statistics Netherlands, a person is a simple unit. For a doctor, a person could be a composite unit, such as when the doctor considers a person as a system of organs. Whether a unit is considered as single or composite depends on the matching problem in question. Antonym: composite unit.
Soundex algorithm	Originally a phonetic algorithm to index names based on sound (in English). Later, a similar algorithm was developed for words in the Dutch language. Improvements of the Soundex algorithm for English include Metaphone and Double Metaphone.
Statistical matching	Matching records with information from units which do not necessarily have to be the same, but are similar. In terms of intention, this method deals with an entirely different problem than is discussed in this report. This is actually an imputation method. This method is not further discussed in this report for this reason.
Surjection	A function $f : X \to Y$ is a surjection if for each $y \in Y$ there is an $x \in X$, such that $y = f(x)$. This type of function is also called surjective.
Synthetic matching	See: Statistical matching
Type I error	See: Mismatch
Type II error	See: Missed match
UWV	The Dutch Employee Insurance Agency (*Uitvoeringsinstituut WerknemersVerzekeringen*).
Weight	See: Matching weight

2. Overview of the matching problem

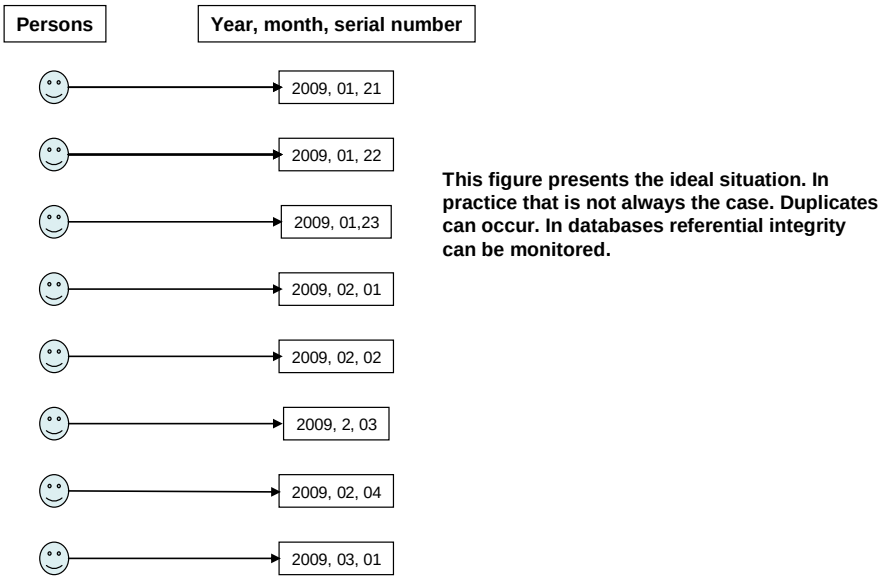
2.1 What is matching?

Matching is bringing together the information from two or more records, which are believed to relate to the same unit, 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.

The matching takes place in *two steps*:

1. It is determined which records are *matching candidates*, 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.

Figure 2.1 Composite primary key



Chapter 4 takes a more detailed look at both steps and the requirements that are placed on feasible solutions, from which the best choice should be determined.

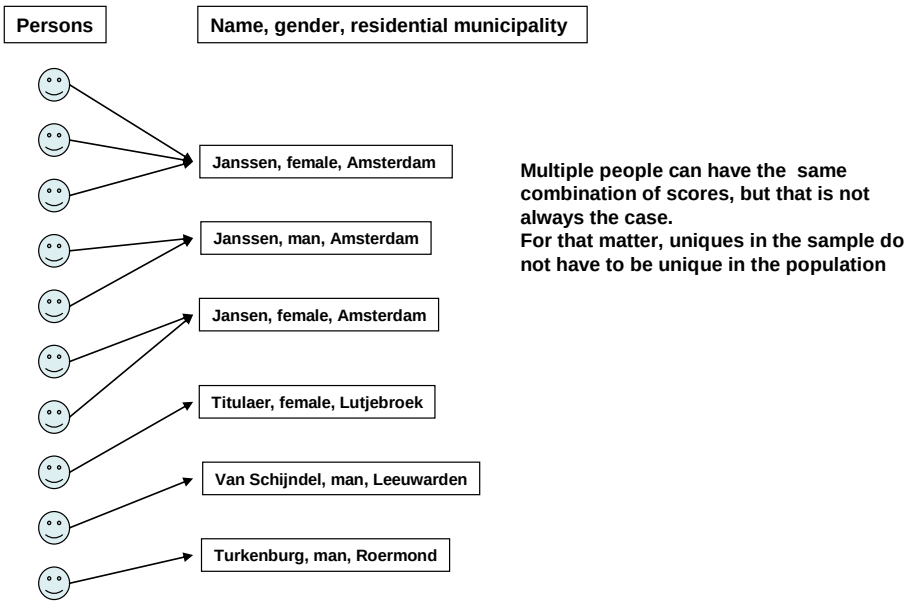
In this document, we discuss two *groups* of matching *methods*.[3] In the first group a consideration takes place in the first phase of the matching process as to which records are matching candidates. This consideration is based on a matching criterion in the form of a decision rule, For this purpose,

[3] Strictly speaking, we consider joining described in Chapter 5 not as a method, but instead as a procedure (in the terminology of Van de Laar, 2008) because this is yields exact results. This is in contrast to methods that are used to find approximations.

a matching key is used consisting of several (key) variables that both files have in common. For example, the matching criterion can then be: 'exactly the same scores on the matching key'. Sometimes, this criterion can be too strict, because errors also may occur in the scores of the matching keys of the files. A weaker form of this matching criterion can offer a solution. For example, if a matching criterion for multiple matching variables coul be: 'exactly the same scores on at least *m* of the *n* matching variables'. Here, n is a given parameter and m, where $0 < m < n$, is a parameter to be established. In the second group of methods, a matching weight to be calculated is used to indicate the extent to which two records match.

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

Figure 2.2 Composite secondary key



The records to be matched can be identified by a single variable or a set of variables. These so-called matching variables together form the identifying key of the record or the unit, the primary matching key (or primary composite matching key). See Figure 2.1. Primary matching keys are unambiguous and, at least in theory, duplicates do not occur. However, in practice that is not always the case. There can also be matches based on other variables in the record, the so-called secondary keys. Such key variables can also be used to identify units, but they are not as hard and also not designed to unambiguously establish units. The possibility of duplicates occurring

therefore cannot be excluded. Nonetheless, several such secondary keys can often be used effectively to identify and match units.[4] See Figure 2.2.

Foreign keys are also used in databases. A foreign key itself does not identify the record concerned, but it is a reference or link to another table in which the key concerned does occur as the primary key. For example, to match a record of an employee, identified by a staff ID number, with data about the enterprise, identified by a unique identifier (BEID), where the employee works. In the table with employees, for each employee record, a BEID is present as a foreign key that uniquely links the table with the entreprise details, where the BEID is the primary key. The condition for this is that foreign key value also actually exists, otherwise the reference will be to a unit that does not exist. In databases, this characteristic is referred to as 'referential integrity'.

2.2 What makes matching so complex?

At first glance, the matching of files seems to be a simple task. In practice, however, this is seldom the case. The following causes contribute to the fact that files are not always easy to match one to one:

- 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, does 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 individual people and a file with households. Or one 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. A foreign key may also be used for this purpose;
- 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, which means that they can be converted to the same denominator. See Section 7.3.1.2 for a further discussion of this problem;
- 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 entreprise 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

[4] These concepts are also used in statistical security. As a rule, primary keys are not present in secured files in that context. The question therefore is whether the files are sufficiently secure, in view of the secondary keys that are present in the files.

enterprise may not be the same anymore. Another example is a match based on residential address, which has new residents because the former residents have moved.

The differences in the files to be matched can make the matching complex and can lead to the following two types of errors:

- Records that are matched, but are not actually associated with the same unit (*mismatch, false positive match, Type I error*);
- Records that are not matched, but are actually associated with the same unit (*missed match, false negative match, Type II error*).[5]

Finally, these types of errors can also be introduced by the choices that are made in the matching process itself. For instance, an incorrect or overly limited matching key may be used, the way in which the weights are calculated may be incorrect, or the cut-off values against which the weights are set off may lead to matching errors.

Table 2.1: Possible errors in the matching/non-matching of two records

	The records are associated with the same unit	**The records are not associated with the same unit**
The records are matched	- good result - rightly matched	- mismatch - false positive match - type I error - erroneously matched
The records are not matched	- missed match - false negative match - type II error - erroneously not matched	- good result - rightly not matched

2.3 Steps in the matching process

As said, matching in itself often seems relatively simple, but in practice, however, the situation is frequently very different. In particular, a lesser quality of data and a lack of clarity about this can lead to all sorts of problems. Matching cannot be viewed as separate from the pre- and post-processing phase, and it is therefore also important to concentrate on the matching *process*. Four different phases can be distinguished (see Gill, 2001; see also Figure 2.3):

A. Orientation phase:

1. Determining the *objective of the matching.* What should the result of the matching be? Is the goal to realise as many matches as possible with minimal certainty or are we only interested in

[5] This concerns only situations with the same units in both matching files, where effects resulting from the dynamic of the population with which the units are associated are not present or are negligible.

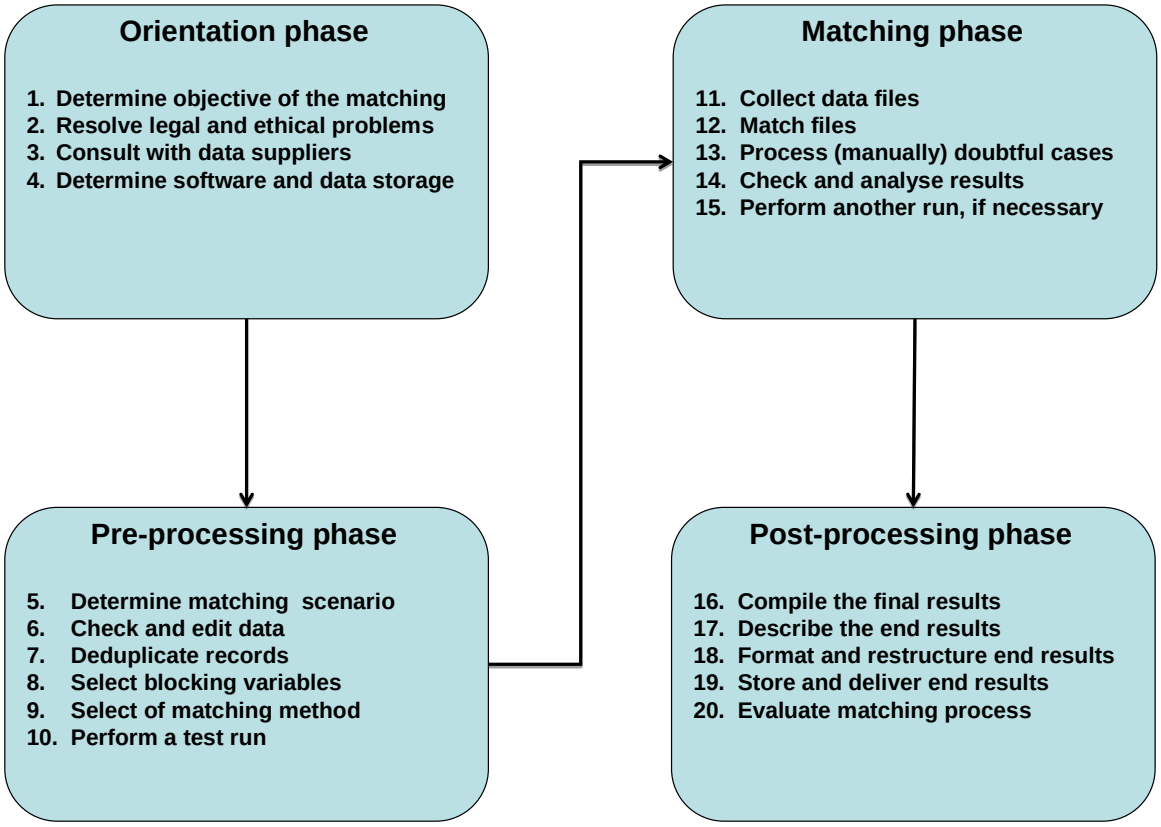
matches with high certainty? How bad is it if a match is missed? Or is it bad if a match is made erroneously? Which matching variables will be used and what is the quality of those matching variables? These types of aspects determine to a significant extent the other steps in the matching process;

2. Resolving *legal and ethical problems*. An initial question is whether there are limitations with respect to privacy, possibly laid down by law. This is the reason why, for the matching of persons, Statistics Netherlands has introduced the RIN number, which replaces the original BSN number for the actual processing and matching. For businesses, it is important to realise that the results of matched files will not always produce a positive response from businesses involved. Because external files are often used, agreements are also needed with the data supplier. It is not self-evident that the results of matched files can be made freely available to everyone (for example, external researchers). That applies in particular if microdata is involved. Another question is how the physical and other security of the data is provided for.
3. Consulting with external or other *data suppliers* and acquiring the files to be matched. Two issues play a role here. The first concerns the way in which the files are delivered and what is delivered. Consider, for example, information about the population and the meaning of variables (including the domain), the format and the structure in which the data is delivered. The periodic delivery of files to be matched should ultimately lead to good agreements between the parties involved, and these agreements should be set out in a Service Level Agreement (SLA). The second, equally important, aspect concerns the acquisition of as much as possible information from the data supplier about the data in the file itself. This concerns information about for example how the data was obtained and processed, how the observation took place, whether edits were performed – and if so, which ones –, whether any strange constructions were utilised, for example, by using fields in a file for purposes other than which they were intended, and how the quality of the data – primarily the quality of the matching variables – should be assessed. Knowledge about these types of issues can significantly reduce the amount of work needed at a later phase in the matching process. Developing this necessary knowledge has proven to be very labour-intensive in practice. Once this knowledge has been obtained, be sure to record it. That is especially important if there is a lot of staff turnover.
4. Determining the *software* to be used *and the storage* of the interim and final results. Statistics Netherlands uses the matching software TRILLIUM, its own customised programmes, and packages such as MS Access and SPSS. The question is which software works best with the specific matching problem. For the storage of the interim and final results, you must consider what interim storage points (ISPs) should be used: are local ISPs available or should these be developed, or can the interim and final results be stored in the Data Service Center (*DSC*), for general use by other parties?

B. Pre-processing phase:

5. Determining the *matching scenario*. First of all, this concerns choosing which matching variables to use, and establishing or assessing their quality. You can then also specify the preconditions that will be used for the matching, such as whether 1:1 matches will be used, or whether a record in one file can be matched with more than one record in the other file.

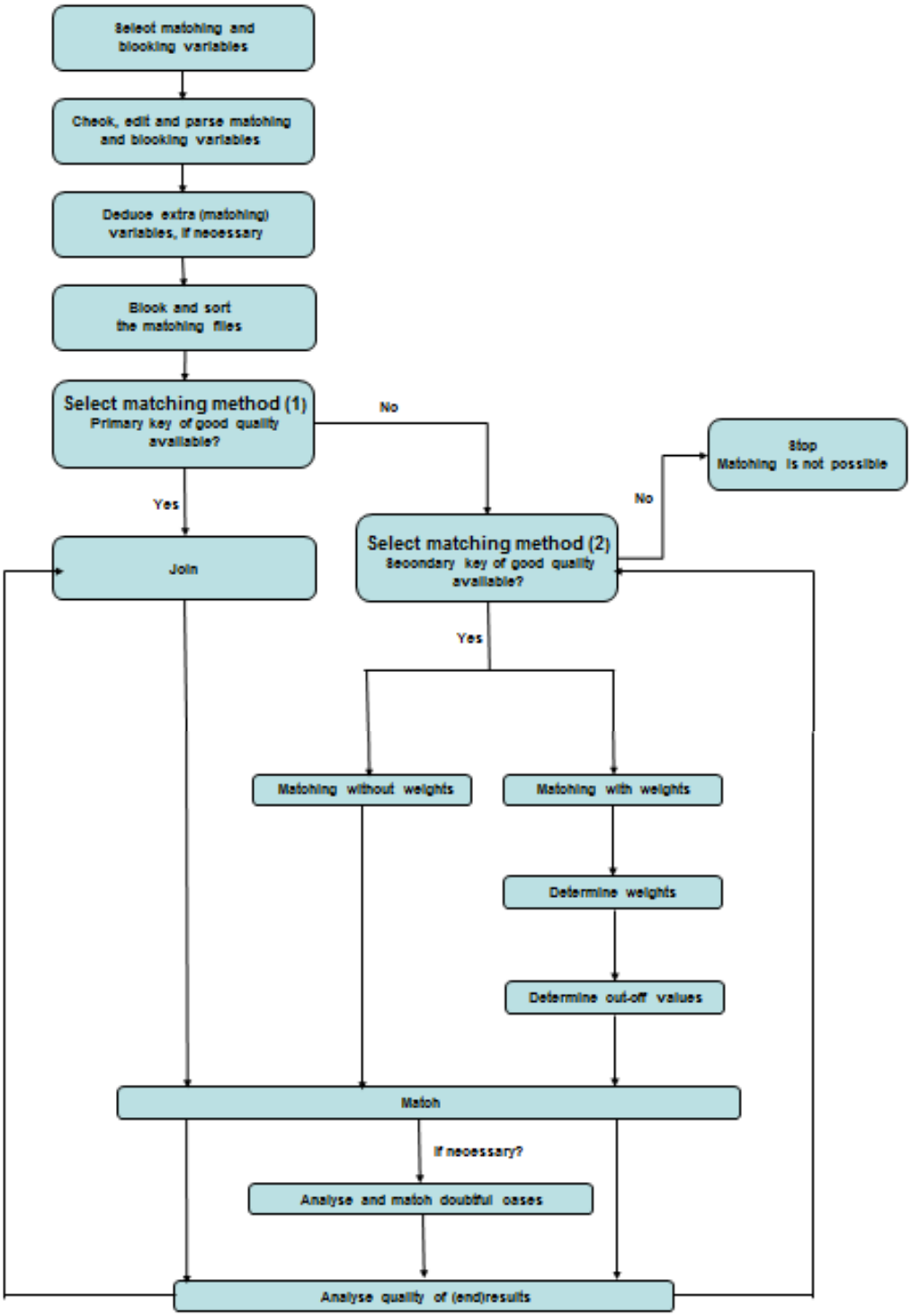
Figure 2.3 Phases in the file matching process



6. *Checking and editing* the matching variables to improve the quality and the structure. The lack of 'noise'-free data is often the biggest obstacle to data matching. This concerns converting raw data to more standardised and consistent data in a form that is suitable for matching. For instance *parsing* variables, in which larger, frequently free-format, strings of variables are subdivided into their components and standardised as much as possible, so that they can be more easily processed by the computer and compared. Consider, for example, separating the two parts of a Dutch postal code, the numeric part and the letter part. Other examples including standardising addresses and names, 'filling up' missing values or simple checks of spelling errors and the value range. Another aspect is standardising classifications over the records to be matched. A large number of varied techniques have been developed for all these types of operations. See, for example, Herzog et al. (2007). A final option which is not often mentioned involves better aligning the units or matching keys of the files to be matched. One example is the recent creation of a new Enterprise Group in business statistics, by which it is better possible to match with the units of the Dutch Tax Administration.
7. Sometimes, it is necessary to investigate whether *duplicate records* (duplicates) are present in a file, and to remove these if this the case ('deduplication');
8. Possibly selecting *blocking variables*. For example, two files to be matched from, each containing 1000 records, produce 1,000,000 potential matching candidates. Therefore, checking all the matching candidates for a possible match is very inefficient. In practice, files to be matched are often many times larger and, for this reason, blocking variables are often used. These variables divide (partition) the files to be matched into two or more blocks, in

which records are compared. If the quality of the selected blocking variable is not too high, then several runs are performed with different variables serving as the blocking variable. It is important to realise that selecting the blocking variable is not self-evident. An 'incorrect' choice can, for example, result in poor end results;

Figure 2.4 Steps in the matching process: pre-processing phase and matching phase



9. Selecting the *matching method*. See Chapters 5 to 7;
10. Performing, analysing and describing a *test run*, if necessary (see further C. Matching phase). This applies mainly if a more advanced method is used, for example, with matching weights and cut-off values.

C. Matching phase:

11. Collecting the *data files* (which may have been adapted previously);
12. *The matching itself. The following steps can be distinguised*:
    - Sorting the files, if necessary;
    - Determining the *weights and cut-off values*, if necessary;
    - *Matching* the potential records to be matched based on the matching key (*the set of matching candidates*), as the first main step;
    - *Comparing the different matching candidates and deciding if there is a 'real match' or not* or whether it is a doubtful case, as the second main step;
    - *Saving the above in special files*.
13. *Manually processing* the doubtful cases, if necessary;
14. *Checking and analysing* the match result and determining the *quality indicators* (Type I and Type II errors). An option is, for example, to take a small sample from the end result and to check it manually to see if it is correct or not. The quality measure can then be calculated on this basis;
15. *Performing the run again*, if necessary, with, for example, other blocking variables, weights and cut-off values or applying the conditions less strictly in the comparison.

D. Post-processing phase:

16. *Compiling the final result* (with files of matched and unmatched units);
17. *Describing the end result*, such as by means of quality indicators, the process performed and the methods used (with parameters);
18. *Formatting and/or restructuring the end results differently* for delivery or storage, if necessary;
19. *Storing or delivering the end results* to the client or for the following step in the process;
20. Performing an *evaluation* of the process undergone so that lessons can be learned for the following cycle.

3. Graphs and metrics for matching

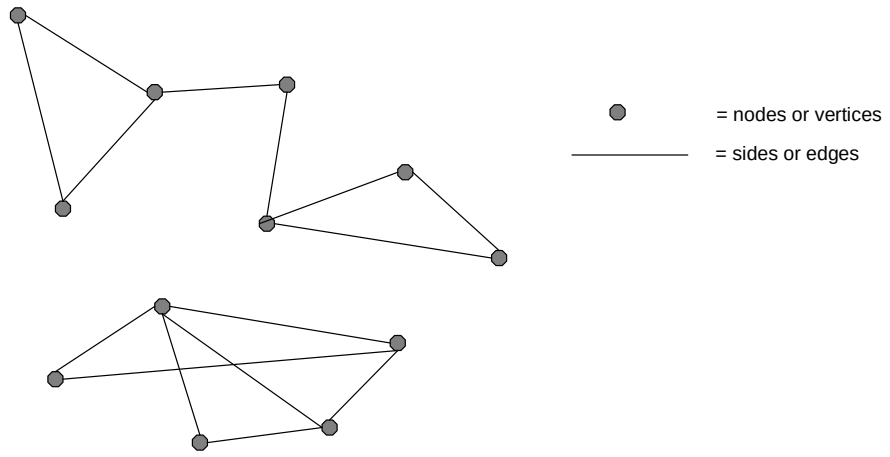
This report uses *graph theory* to describe the theory behind matching and the classification of matching methods. For this reason, this chapter takes a brief look at graph theory prior to the discussion about the theory and the matching methods. In particular, several basic concepts are explained that are important for the text that follows.

This chapter also examines another concept important for this document, the concept of *metrics*. A metric is used here primarily to determine the matching weights. The subject is introduced in this section, and is discussed in more detail in Chapter 7.

3.1 Graphs

A graph G ($G = (V, E)$) consists of a finite set of points V, also called nodes or vertices, of which some pairs are connected by lines (E), also called sides, edges or branches. A graph is depicted in Figure 3.1.

Figure 3.1 Example of a graph

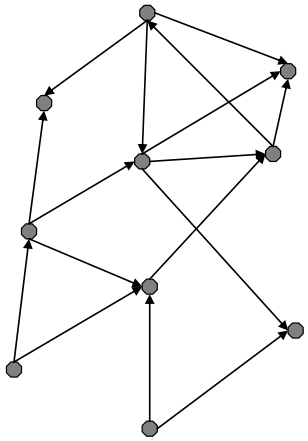


Depending on the intended use, the lines can be directed and then they are called arrows. In that case, we refer to a directed graph or a *digraph*, an abbreviation of 'directed graph'. See Figure 3.2. Weights can be assigned to the lines in the form of real numbers.

A graph with weights associated with points or edges is called a *weighted graph*. In this document, the weights are associated with the edges, and they express the strength of the match between two records. There are different ways to calculate these weights. In this document, these weights represent matching weights that show the strength of potential matches between records. You can

also see the weights as distances: the smaller the distance, the more similar the keys of the records are.

Figure 3.2 Example of a digraph



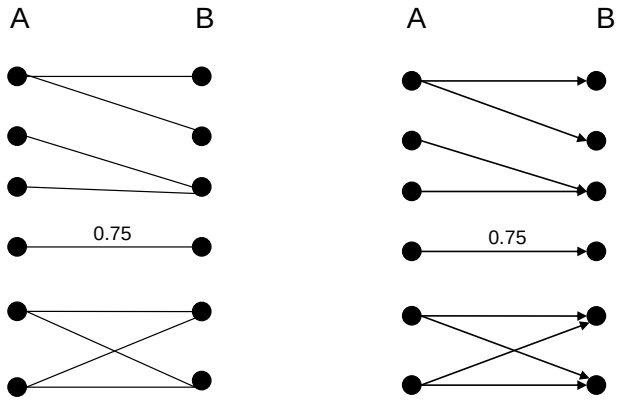
A special type of graph $G = (V, E)$ is the *bipartite graph*. See Figure 3.3. Here, the set of nodes V can be divided into two disjoint sets A and B. Therefore the following is true: $V = A \cup B$ and $A \cap B = \phi$. The edges exclusively connect nodes in A with nodes in B. There are no edges in A and B individually. In this context, it is permissible that one of the two sets – or even both sets – are empty. The bipartite graph is highly suited to illustrating matching and the theory behind it.

Finally, this document discusses the *MC graph,* the *matching candidate graph*. This is a bipartite graph that represents the possible matches between records from two files. The edges may or may not be assigned matching weights. A matching candidate graph symbolises part of the constraints that apply for a matching problem.

Notation:

A graph G is denoted as a pair (V, E), where V represents the set of points (nodes or vertices) and E the set of lines or edges. Each edge e in E is a set $\{a, b\}$ where $a, b \in V$. A directed graph, or digraph, is a pair $(V, \overline{E})$ where V is the set of points and $\overline{E}$ is the set of arrows. In this context, an arrow $\alpha \in \overline{E}$ is an ordered pair $(a, b) \in V \times V$. The $|.|$ denotes the function that shows the number of elements of a set (possibly, ∞ = infinite). In this document, all graphs and digraphs are finite, which means that the number of points is finite, therefore $|V| < \infty$, and therefore also the number of edges (in graphs) or arrows (in digraphs) is finite.

Figure 3.3 Two examples of a bipartite graph



In both examples the points are two disjoint sets A and B. The edges exclusively connect the nodes in A with nodes in B. An edge can have a weight (number >0).

The arrows in the second example are all directed from nodes in A to nodes in B. In a general bipartite digraph the arrows can point from nodes in A to nodes in B, as well as vice versa.

Number of connectivity components: 4.

Paths and connections in a graph:

A *path* in a graph is a succession of nodes arranged in such a way that an edge runs from each node to the following node in the row. Given a graph $G = (V, E)$ where v and w are two points of G, so $v, w \in V$. A path in G from v to w is a sequence $v_1, ..., v_k$ of points in G, such that:

1. $v_1 = v$
2. $v_2 = w$,
3. $\{v_i, v_{i+1}\} \in E$ for all $i = 1, ..., k-1$.

If there is a path from v to w in G, then there is also one from w to v (symmetry). If there is a path in G from u to v and from v to w, then there is also one from u to w (transitivity). Here, u, v and w are points in G. For each point v in G, there is – by definition – a path from v to v (reflexivity). In other words, the relationship 'connected by a path in a given graph' is an *equivalence relationship* to the set of points of the graph, i.e. a binary relationship that is reflexive, symmetrical and transitive. If there is only once equivalence class for a graph G, G is said to be *connected*. In that case, all pairs of points can therefore be connected with each other by paths in G. If there are two or more equivalence classes for a graph, G is said to be *non-connected*. In that case, an equivalence class of this relationship corresponds with a *connected component* of G; this is a connected subgraph of G. See, for example, Figure 3.3.

Suppose $G = (V, E)$ is a graph, and that v is a point in G, so $v \in V$. The degree of v (in G) is the number of edges $e \in E$ for which $v \in e$, so the number of edges in G on which v lies.

3.2 Metrics

This document makes use of *metrics*. A metric is a function that defines the distance between each pair of elements of a set. Sometimes, it concerns a function that is related to that of a metric, but which deviates on several components from that of a metric. In that case, we have generalised metrics. But we will discuss metrics here first.

We assume a set X for which function $d : X \times X \rightarrow [0,\infty)$ is defined that satisfies a number of conditions.

1. $d(x, y) = 0$ if and only if $x = y$,
2. $d(x, y) = d(y, x)$ for all x, y in X (*symmetry*), and
3. $d(x, z) \leq d(x, y) + d(y, z)$ for all x, y, z in X (*triangle inequality*).

A non-negative function d that satisfies conditions 1, 2 and 3, is called a metric. Sometimes, there is a stronger condition instead of attribute 3:

4. $d(x, z) \leq \max\{d(x, y), d(y, z)\}$ for all x, y, z in X

A non-negative function d that satisfies 1, 2, and 4 is called an *ultra-metric*.

In practice, there are also functions with properties that deviate from those of a metric. In some cases the range (codomain) of d is equal to $[0,\infty]$, or it is not true that $d(x, y) = 0$ implies that $x = y$, or the symmetry attribute does not apply for all pairs $x, y \in X$. These types of functions are generally designated as generalised metrics, and specific names are also used, such as pseudo-metric, quasi-metric, semi-metric, hemi-metric or similarity measure/dissimilarity measure. In statistics, they are used in, for example, cluster analysis (see Mardia et al., 1982).

In matching and specifically in the comparison of matching keys, this concerns the measurement of the distances between the scores for the matching keys, or, in other words, determining the comparability or non-comparability.

In general, we denote by d, d_H or $d(.,.)$ a metric. We denote the scores on a matching key as a vector $(\alpha_1,...,\alpha_n)$ for a matching key $(v_1,...,v_n)$.

A few of the metrics we use here are so special that they are specified separately. Here, we mention the *Hamming distance*, denoted by d_H. It is defined by:

$$d_H(\alpha, \beta) = d_H((\alpha_1,...,\alpha_n),(\beta_1,...,\beta_n)) = |\{i \mid \alpha_i \neq \beta_i, 1,...,n\}|, \qquad (3.2.1)$$

i.e. the number of places in which the vectors α en β have different scores. Note that, in principle, the Hamming distance can be defined for all types of variables.[6] An example: suppose that there are two matching keys of four alphanumeric figures, '1034' and '1135' respectively. In this case, the

[6] Incidentally, this does not mean that the metric would always be a preferred one. For an alphanumeric variable, for example, such as a surname, the tendency will be to choose a metric that will make a gradual distinction between different names. For example, 'Jansen' only differs by one letter from 'Janssen', but by six letters from 'Boog'. The Hamming distance only records that both names are different.

Hamming distance is 2, because the figures differ in two places, which are positions 2 and 4. In other words: the smaller the Hamming distance the greater the comparability of the matching keys. The Hamming distance is equal to the number of 'changes' that must be made in one key value to obtain the other key value.

Another metric that we want to explicitly point out here is that of *Levenshtein*, denoted as d_L. This works on strings and counts the number of elementary operations, such as deleting, adapting or even adding characters, needed to transform one string into the other. In contrast to the Hamming distance, which can be called a universal metric, in the sense of: being applicable for every type of variable, the Levenshtein distance is a metric that was specifically designed for comparing strings. There are more metrics of this type, specifically tailored to a certain type of variable, as shown in Section 7.3.1. Example: the Levenshtein distance between the words '*farce'* and *'pursue*' is 3: 1) *farce* becomes *parse* (f replaced by p), 2) *parse* becomes *purse* (a replaced by u) and 3) *purse* becomes *pursue* (u is added). The advantage of the Levenshtein distance, compared to the Hamming distance, is that it can process key values of different lengths.

The following is a special case of a metric for a matching key consisting of multiple variables. We could, for example, have a matching key that consists of n variables (all secondary key variables), all of different types, and for which the i^e variable has a metric d_i. For the entire matching key, we can define a metric d by *adding the metrics of the separate variables of the key in a weighted manner*, where, for each weight w_i, $w_i > 0$, $i = 1,...,n$. We then obtain $d = \sum_i w_i d_i$. For that matter, the weights are needed to align the separate submetrics with one another. In this context, use is made of the fact that if δ is a metric, $a\delta$ is also a metric for each $a > 0$. By using the weights, we can align the metrics d_i with one another. Incidentally, it is not necessary to always use a single metric if we have a matching key consisting of multiple variables. We could also work with the metrics for the separate variables.

In some cases, an indicator vector is needed, which indicates in which places α en β differ. Be δ a 0-1-indicator function, which is defined as follows: $\delta(a,b) = 0$ if $a = b$ and $\delta(a,b) = 1$ if $a \neq b$, for scores a, b for a matching or other variable. For score vectors α, β, we define

$$\Delta(\alpha, \beta) = (\delta(\alpha_1, \beta_1),..., \delta(\alpha_n, \beta_n)) \in \{0,1\}^n .$$

This indicator vector plays a central role in the Fellegi-Sunter method (1969) (see Appendix A). Note that $d_H(\alpha, \beta) = \sum_{i=1}^{n} \delta(\alpha_i, \beta_i)$.

4. Matching theory

4.1 Introduction

Suppose that we have matching files A and B, and that these files contain information relating to periods that are not too far apart in time. The following possibilities exist with regard to the matchability of these files (see also Figure 2.4):

1. There is a common and unique primary matching key that is present both in file A and file B. Two possibilities follow from this:
   a. The scores on the variables in the matching key are of sufficient quality.
   b. The scores on the variables in the matching key are of insufficient quality.
2. There is no good common and unique primary matching key present in both files. However, the files contain certain common variables that could serve as a secondary matching key. There are two possibilities in this case as well:
   a. The scores on this common secondary matching key are of sufficient quality.
   b. The scores on this common secondary matching key are of insufficient quality.

It is clear that this is a summation of types of matching problems, listed in order from easy (case 1a) to difficult or even impossible (case 2b). The difficult matching cases are those that fall under 1b or 2a. These are the matching problems that this document will focus the most attention on. For the sake of completeness, situation 1a will also be discussed (Chapter 5), but no methodological problems play a role here. In the terminology of Van de Laar (2008), this concerns a procedure and not a method.[7]

Matching criteria and preconditions:
The use of a *matching criterion* produces records that can potentially be matched, the so-called matching candidates. These matching candidates are determined first, in the case that matching methods are used, which means in situations with secondary keys and errors or deviations in the data. For example, if you use a metric to measure the extent of similarity between two records, then the matching criterion indicates the distances (such as cut-off values) that you should stop at to still be able use two records as matching candidates. Suppose that you have five secondary matching variables as a composite key. The matching criterion could then be that records that have the same score on at least three of the five matching variables will be considered as matching candidates, while the rest will not.

To match records in two matching files, we need more than just variables in both files to perform the match. We also need to know the *preconditions* under which the matching must take place.

[7] The difference is that a method concerns an approximation, and a procedure does not. For example, weighting is needed to obtain approximations for population figures from population estimates. This type of weighting is based on one of the many known weighting methods, which are each suited to specific situations. Weighting provides estimates of population sizes. When matching two files based on a hard key (a primary key), there is no approximation. (See Chapter 5) The matching methods based on secondary keys (see Chapters 6 and 7) are approximations.

Usually, a match is performed so that no single record in both matching files may be matched with more than one record from the other file (1:1 matches), and it is also possible that records will not be matched. However, there are also situations in which other preconditions apply. For example, there could be a situation in which each record from a file A must be linked with at least one record from another file B, while each record from B may not be matched with more than one record from A (1:n matches). This requires that file B has at least the same number of records as file A. This situation can arise if the units in A form a subset of the units in B. In the case, for example, that we must take account of splits and mergers of units, the preconditions are different yet again. In that case, it must be permissible that multiple units from file A can be matched with one or more units from B, and vice versa (m:n matches). For a matching problem, it is important to know exactly under which preconditions a match will be made.

Method:

To select a matching method, you must know what type of match you want to perform, either without matching weights, or with them. In the first case, the potential matches that are found all count the same in terms of weight. This is not true in the second case, where weights can be used to indicate how strong a candidate match is.[8] These matching weights can be calculated in various ways. For example, you can use metrics, similarity or dissimilarity measures, probability models, etc. Section 7.3.1 takes a more detailed look at calculating these weights.

Target function:

A certain class of matching problems (with matching weights) uses a target function that must be optimised (minimised) under certain preconditions. In the target function, the different matching candidates are given a weight: the matching weight. This matching weight is used to make a slight differentiation among the strength of potential matches. As indicated above, there are various ways to calculate matching weights.

Specific situations:

There may be situations where, at first glance, it seems like no good primary or secondary matching key is available, especially if some of the variables of these keys do not have exactly the same domain. Example: one file has an age variable with the age in years as its domain (such as the set $\{0,1,2,...,120\}$), while the other file has an age classification in five-year classes (such as the set $\{0-4,5-9,...,100+\}$. In these types of situations, these variables can still be used as matching variables. The techniques that can be used for this are not much different than in the case of situations 1b of 2a.

Another specific situation concerns the fact that the condition is not satisfied that the scores in both files should relate to approximately the same point in time, therefore with about the same reference times. It is possible that the points in time[9] to which the data in these files relates are so far apart

[8] Formally, the methods that do not use matching weights are a special case of methods that do use these weights. After all, you can associate weights with all candidate matches that have the same value, for instance 1. Because the solution methods are different for the two types of matching methods, we do make this distinction here (just as it is done in the literature for combinatorial optimisation).

[9] This concerns the point in time to which the data relates. However, even if the point in time to which the two sets of data relate is the same, there can also be a big time difference in terms of when the data was recorded. The matching problems referred to can also arise in this case.

that differences in the scores of the same units arise purely because of dynamics in the population, which means that there may be an influx of new units in the population ('births'), or an outflow ('deaths'), or the attribute of these units may change (for example, turning a year older, getting married or divorced, etc.). In addition, the units themselves may also change. This is possible, for example, for composite units such as businesses or households, which may split or merge with other units.

Matching is usually used to link records from two files 1:1. This means that, in the definitive matching, if two records match, r_A from A and r_B from B, there are no records s from A and t from B, such that r_A matches with t or r_B with s. However, it is still very well possible that separate records from A can be matched with multiple records from B, or vice versa, separate records from B with multiple records in A. Consider matching situations in which there is such a difference between the two measurement times when A and B were collected that the effects of the dynamics become visible. For example, an enterprise from file A could have split into several enterprises represented in file B. Or, conversely, an enterprise in B could have arisen due to the merger of several enterprises in A. This does not necessarily have to involve composite units such as businesses; it may also involve people. For example, a person who was 32 years of age in file A could be 32 or 33 years of age in file B if the data on this person were collected later in file B. If there are two people in B that have the same scores for secondary key values on all matching variables, but only a different score on the variable of age, namely 32 years of age and 33 years of age, then both people in B are matching candidates for the stated record in A (assuming that no further identifying information is present in A and B). With a certain probability, both records from B can be matched with the record in A. Here, the reciprocal of the matching probability can be used as the matching weight. This is an example of a matching model that uses matching weights. However, this is not always necessary. There are also matching models that do not use such weights.

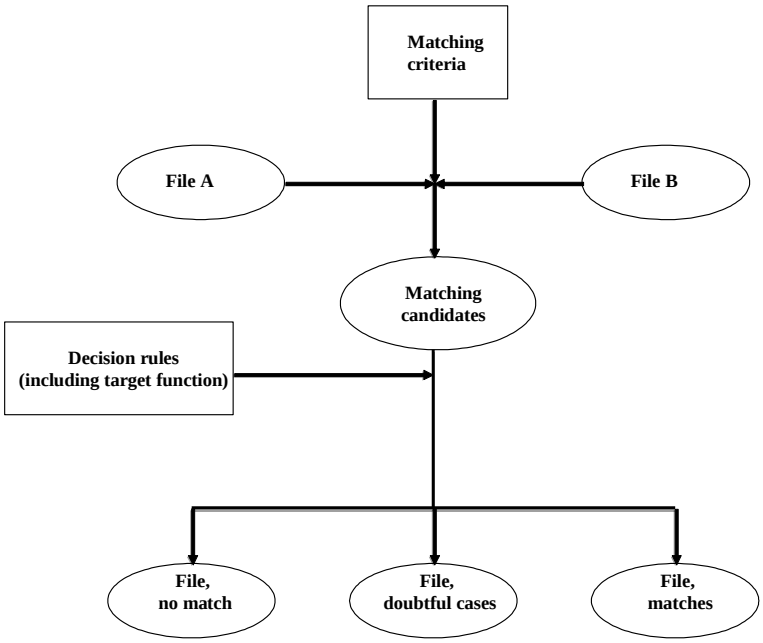
This document also examines more than just matches of identical units. Economic statistics often deal with composite units, which can split, or merge with other similar units into a new unit. See also Section 9.6. The relationship between two units in different files to be matched does not necessarily have to be that of the sameness of units, but, for example, that of 'arose from' (in the event of a split) or, conversely, 'became a part of' (in the event of a merger).

Finally, in matching, it is important to know whether all records from one file must be matched with records from the second file, or only with some. And also whether we want to match as many records as possible with a correspondingly high risk of creating mismatches, or conversely, if we instead intend to make only high quality matches, and take for granted that some (correct) matches will be missed.

We have now discussed all the ingredients of a matching method.[10] In the rest of this chapter, we will discuss and explain these elements in more detail, to prepare for the extensive discussions in Chapters 5, 6 and 7.

[10] This means mainly the more difficult matching models 1b and 2a. The situation in 1a is trivial and is not covered here.

Figure 4.1 Main ingredients in matching



4.2 Choosing between the matching methods

Matching involves creating relationships between data from different files. A matching criterion is used for this purpose, and possibly also a target function and preconditions that a permitted match must satisfy. You can also look at the intention behind matching this data, and in particular, to which population units it relates. If it relates to the same units, then it concerns one of the matching techniques discussed in this document. However, if it relates to similar, but not necessarily the same, units, then it concerns *statistical matching*, also known as *synthetic matching.* As indicated, this class of techniques falls outside the scope of this document, because these methods are rather the domain of imputation methods as far as their intentions and execution concerns.

In this document, we examine three matching methods:

- Matching based on a – possibly composite – primary key ('joining');
- Matching based on a – possibly composite – secondary key, not using matching weights;
- Matching based on a – possibly composite – secondary key, using matching weights.

Matching based on a primary key is actually the ideal matching method, because here the units are identified unambiguously and uniquely. In theory, no duplicates can arise. In practice, however, some errors will creep in. For example, a single file could (erroneously) have duplicate records.[11] In

[11] For that matter, the duplication of records with the same primary key is not wrong by definition. Consider, for example, a job file or a vehicle file. A person could have several jobs, or own multiple vehicles. This then concerns foreign keys. In the person file, each person must have an unambiguous person number (the BSN). However, there are also situations in which the separate records in a file are considered to relate to different

that case, deduplication must first be performed. If many duplicates are present, we are dealing with a matching key that is not very reliable. In that case, it may be a better idea to look for alternatives, in the form of secondary keys in the matching files.

If the two matching files do not have a common primary key, but a common secondary matching key, then this can be used to match records. However, here we must consider that several matching errors are possible: matches can be made while they are not correct, or they can be missed unintentionally. We look at two options for this situation: matching weights are either used or not used to indicate the strength of a possible match.

Which method is selected depends to a large extent on the situation. Aspects to be considered are:

- *The quality of the matching keys*. If the quality of the matching variables is good and they are strong identifiers, we are more apt to choose 'joining' (i.e. if a good unique primary key is present) or a method without weights;
- *The unambiguousness and comparability of the matching keys*. If the matching keys are strongly discriminating, then the tendency is to use methods without weights instead of a method with weights;
- *The desired quality of the matches*. The methods without matching weights – in general terms – perform better but offer lower quality, while models with matching weights produce better quality but do not perform as well;
- *The available hardware and software*. Much of the matching software does not allow for the proper use of weights, and as stated above, 'joining' and a method without weights will perform better than a method with weights;
- *The time, available capacity and knowledge*. If these elements are limited, then one will tend to choose 'joining' (if a good primary key is present) or a method without weights, because these are easier to perform than a method with weights, and less knowledge is required. It should be noted that such a choice does not have to be optimal in view of the quality and comparability of the matching keys. A method with weights initially requires a lot of time: not only do the weights have to be determined (and the weighting method), but also the correct level of the cut-off values, above or below which matching candidates are still seen – or not – as true matches. To properly determine these weights and values and to get a feel for the process, several runs of the match are needed. An advantage of a method with weights is that, by raising or lowering cut-off values, it is possible to experiment with the number of matches which are seen as doubtful, and therefore with the capacity that is necessary for manual processing.

4.3 Matching models based on graphs

In this section, we describe matching from a theoretical perspective and explain how modelling the matching problem works: the problem formulation and the solution. In practice, certain parameters must be filled in, tailored to the specific matching situation. Matching weights are an example of this.

---

units, such as people. In such a case, all records in the file must have a different key value. The presence of two records with the same key value is then an error.

The discussion of the matching methods in this chapter focuses on the so-called MC graph[12] when matching two files. This is a bipartite graph, in which one set of points represents the records in the first matching file, say file A, and the other set of points represents the records in the other matching file, say file B. The edges in the MC graph represent potential matches.

As indicated before, in this document we make a distinction between two groups of matching methods (apart from 'joining'): methods without weights and methods with weights. The first group of methods works with an MC graph in which all edges are considered as equivalent. This is not so for the second group of methods. The differences between the edges in the MC graph are expressed using matching weights. This chapter briefly examines the different methods that exist to create matching weights. An important subclass of matching methods that belongs to the second group is formed by probabilistic matching methods. The fact that we need such techniques has to do with uncertainty in the matching due to errors or irregularities in the data, the use of slightly different matching variables or population dynamics, as a result of which unit attributes can change over time, but not always in an unambiguous way. For example, after a certain period, an neterrpise may exist as it has been for some time, or it may have gone bankrupt, or merged, or have been acquired by another enterprise, and all these possibilities with different probabilities.

In practice, matching usually involves large MC graphs, which will have relatively few cases of matching pairs of which one of the points is present in multiple potential matching pairs. The problem (and the work) in matching therefore lies mainly in calculating candidate matches and the possibly associated weights, and not so much in making a selection from alternative candidate matches, because these are expected to occur relatively rarely. Furthermore, this will often concern relatively small selection problems that can all be resolved separately. Each of these selection problems corresponds with a connected component of the associated MC graph.

Another point is that, in statistics, cases with alternative matches are seen as doubtful cases, which must be presented to a matching specialist for resolution. However, this is often not necessary, and a programme could make the decision. That would involve a lot less work, and it could also lead to better process documentation and audit trailing. Only real problem cases – no more than a handful – would then be submitted to a matching specialist.

Before we concentrate on the matching problems in bipartite graphs and digraphs, we will first take a look at the matching problem in arbitrary graphs and digraphs.

4.4 Matching problems in graphs

Matching can be described with the help of a special type of graphs: bipartite graphs. However, we can also talk about matching in a general graph $G = (V, E)$. The simplest case is 1-matching (also just called 'matching'). The goal here is to choose a subset $F \subseteq E$ in such a way that each point $v \in V$ is on no more than one edge in F. This can also be formulated as: for $G(F) = (V, F)$, the degree of each point v in $G(F)$ is no higher than 1. Each graph always has one 1-matching, i.e. where $F = \phi$, the graph that consists of points from G and which does not have any edges. The

[12] See Chapter 3 for more information about graphs and metrics.

trick is now, for a given graph $G=(V,E)$ to find a maximum 1-matching $G(F)=(V,F)$, therefore where the number of edges $|F|$ is maximal. This is an *unweighted matching problem*.

1-Matchings can be generalised as h-matchings, where h is a $|V|$-vector of integers, and where h_i is an upper limit for the number of edges on which point $i \in V$ lies. An extra requirement can be made for h-matchings, namely that each edge may not be selected more than once. This is called 0-1 h-matching. Another option is that an edge can be selected multiple times, and this is known as integer h-matching. In this document, we use only 0-1 h-matching.

Weights can also be assigned to the edges, the matching candidates. If w_e is a weight for $e \in E$, then $w(E') = \sum_{e \in E'} w_e$ be the weight for $E' \subseteq E$. The *weighted h-matching problem* involves finding an h-matching of maximum weight. The unweighted 1-matching problem is a special case of this, because here $w_e = 1$ for each $e \in E$.

A formulation of weighted 0-1 (1-)matching as an integer programming problem is as follows:

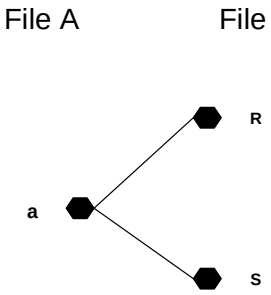
$$\begin{aligned} &\max\ w'x \\ &Ax \le b \qquad (4.4.1) \\ &x \in \{0,1\}^n, \end{aligned}$$

where $A=(a_{ij})$ is the incidence matrix where $a_{ij}=1$ if point j lies on edge i, and $a_{ij}=0$ otherwise. In addition, $b = (\overbrace{1,...,1}^{m})'$, where m is the number of edges, i.e. $m = |E|$. The vector b consists solely of 1s, because we are dealing with a 1-matching. Furthermore, $n = |V|$, the number of points in G. $x_e = 1$ means that edge e is in the matching, and $x_e = 0$ means that it is not.

Matching problems can be formulated as *optimisation problems*. In this document, we let the question of whether a maximum or minimum must be found under restrictions depend on the matching problem in question. Sometimes, it seems more natural to use a maximum, and in other cases, a minimum. In any case, each of these types of problems can easily be converted to the other by placing a minus sign before the target function. For an extensive discussion of matching in combinatorial optimisation, see Nemhauser and Wolsey (1988, Chapter III.2) or Papadimitriou and Steiglitz (1998, Chapters 10 and 11).

A possible approach to resolving the matching problem is as follows. First, we eliminate the records in both files that do not match any records. This produces the first reduction of the MC graph. We can then resolve the matches without alternatives. This creates a further reduction of the problem. If any records remain, these are candidate matches with alternatives. These can be resolved according to the abovementioned combinatorial optimisation method. As a rule, in practice, you will have to deal with a number of smaller and mutually dependent problems from which alternative matches must be selected. If there are many of these, it is best for a program to make these choices (and to record the decisions in a logfile). More difficult cases must then be examined and assessed by matching specialists; simpler cases will probably only be dealt with on a sample basis. Here, ‘difficult’ and ‘simple’ are based on the size of the selection problems to be resolved; the bigger the problem, the more difficult it is.

Figure 4.2 Two possible, equivalent matches



Matches {a,R} en {a,S} are matching candidates. With 1:1 matches only one of the matching candidates must be chosen, meaning the other matching candidate must be declined.

4.5 Working methods

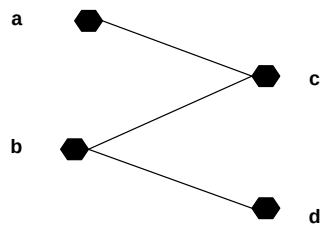
Suppose we have two files that must be matched. First, we must determine which matching key to use. In addition to the matching key, other variables and parameters could be used, such as weights and cut-off values. The objective is now to determine a criterion to match records from file A with records from file B. If this criterion is applied to the records in files A and B, this will produce pairs of records, for which each record satisfies the matching criterion, and which can therefore be matched. Such information can be represented in the form of an MC graph.

Depending on the selected matching method, the strength of the matches may still need to be expressed, in the form of matching weights. A suitable characterisation must first be found in order for this to be determined. Sometimes the matching criterion can be used here, for example, to quantify the extent of deviation from the ideal.

If the MC graph has been compiled, either with or without matching weights, it is then important to formulate a suitable target function that indicates what type of matches we are looking for. Criteria that the solution must satisfy must also be formulated. These types of criteria are usually related to the maximum degree for all points of the matching graph. In many cases, for example, the degree may not exceed 1. This could, for example, be the case if the units to be matched are people. However, there are also situations where a 1:n match is possible. This is the case, for example, when enterprises split: where one enterprise splits into two or more parts, and the parts continue to function independently of each other, that is, with different top management. A match of n:1 is also possible, for example, – in keeping with the enterprise example above – if two (or more) enterprises merge. Even n:m matches are possible, for example, if n enterprises merge, and m new enterprises are created from this total. We will examine similar examples at a later stage in this

document. However, it should be noted that the match in these cases is often interpreted differently. Here, no units that are the same are being matched, but units that originated from other units.

Figure 4.3 Two possible matches



Matching candidates with a 1:1 match: 1. {a,c} and {b,d} and 2. {b,c}

In Figure 4.3, for example, there are two matching candidates available. Assuming the criterion that only 1:1 matches are permitted, there is no choice to be made, unless we can make a selection based on, for example, calculated weights. We find two possible matches here: {a,c} and {b,d}. In principle, there is nothing that prevents {b,c} from being selected as a match. This would certainly be so if we were working with weights, and if the weight for edge {b,c} was larger than weights associated with the edges {a,c} and {b,d} summed.

If {b,c} is chosen as a match, then the possible matches {a,c} and {b,d} will not ultimately be made, under the condition that 1:1 matches must be made.

4.5.1 Matching based on a primary key

Matching based on a unique primary key for which the scores (the key values) in both matching files are also of high quality is, in a certain sense, trivial. These types of matches are performed as a standard operation in database packages. This operation is called a *join*, or more precisely, an *equi-join*.

Matching based on a unique primary key is only useful if the key values are also reasonably reliable, and matches can take place based on the exact similarity of key values.[13] If the values are not reliable, it is not useful to work with a metric and to look for matching candidates that are 'close to', fon instance, social security numbers. The 'closeness' of social security numbers has no relationship whatsoever with the 'closeness' of people in the usual sense of the word.

4.5.2 Matching based on secondary keys, without matching weights

The goal of the first step, model formulation, is to specify a matching problem in the form of an optimisation problem. To do this, several issues must be selected by the person who is responsible for the matching operation. Based on these specifications/choices, some issues must then be derived from the matching files. We will examine each of these components below.

For matching problems without matching weights, the following items and parameters must be specified:

1. The matching key, consisting of the variables from the two matching files that you want to match.
2. The matching criterion that you want to use to calculate matching candidates. This matching criterion applied to the matching files produces an MC graph.
3. For large matching files, it may be necessary to work with a stratification that limits the searching space for finding matching candidates. Blocking variables are used for this purpose. One (or more) blocking variables can be used to calculate this stratification.
4. Degree restrictions that apply for the matching graph. This means that matches must be 1:1, 1:n, m:1 or m:n. It is also possible that m or n must have an upper limit. One record may be matched with multiple records from the other file if the reference dates from the matching files differ.

Once a matching model without matching weights has been specified as an optimisation problem, the next step is to resolve this problem. We discuss solution methods for this type of model in Chapter 6.

4.5.3 Matching based on secondary keys, with matching weights

For matching problems with matching weights, in addition to the four items named in Section 4.5.2, several other items must be specified. These are:

5. The calculation method for the matching weights. These weights can also be calculated when calculating the MC graph.
6. The cut-off [14] value that indicates which matches are considered acceptable. This is a threshold value that ensures that matching weights that are too small (this means that they

[13] In this case, the reference times for the two matching files must not be too far apart. What 'far' means is determined by the dynamics in the population concerned and the matching errors that are deemed acceptable.

[14] Here, it is possible to work with an upper and lower limit at the same time. All matches with weights above the upper limit are seen as true matches. All matches below the lower limit are seen as true mismatches.

are below a lower limit to be specified by the person matching the files) must not be considered as candidate matches. Using these cut-off values, it is possible to influence the risk of missing matches, and also – the opposite – of erroneously making matches.

7. Specification of the target function when using matching weights. As a rule, this is simply the sum of the matching weights of the edges in a feasible matching graph.

We discuss the solution methods for this type of model in Chapter 7.

In this document, we want to define a matching problem – except for a matching problem based on a primary key – as an optimisation problem. In this context, based on criteria that must apply for the solution, a subgraph of the MC graph must be determined that optimises the target function in the problem.

Matches falling in the area between these two limits are the doubtful cases and are presented to the matching specialist. By varying the upper and lower limits, the scope of the number of doubtful cases can be limited or expanded.

5. Matching based on a primary key

5.1 Short description

Matching based on a primary key is the simplest way to match. Both matching files contain the same unique primary key that is used as the matching key. The assumption is that the quality of the primary key is sufficiently high; otherwise this matching method cannot be used effectively. This form of matching is used very frequently, especially because little matching knowledge is required and the method is easy to perform. In addition, this method is supported by many software packages, from Excel and Access to more advanced database and matching packages.

The basic principle is that a match is made if and only if a record from one matching file has exactly the same key value as that of another record from the second matching file. These types of matches are performed as standard in databases, because database management packages contain functionality for this purpose. In database terminology, this involves an operation referred to as a 'join', or an 'equi-join'.

In the sense of Van de Laar (2008), this concerns a procedure and not a method, because there is no approximation involved. However, approximation is involved in the other forms of matching, those based on secondary keys. So this second case concerns methods.

To clarify, if there is a unique primary key, this is usually a key that consists of a single key variable, such as a citizen's identification number (BSN) or a business identification number (BEID). However, this is not necessarily the case. A primary key can also consist of several variables, for example, a key for households with a sequential number for the members in a household. As a key for the people in households, this combination is unique. The fact that the key could be composed of several variables is actually unimportant. For a primary key, the point is that the record is uniquely identified. No duplicates can occur, at least in theory. For secondary keys, which play a central role in Chapters 6 and 7, there are usually multiple, truly different variables that can be used to match units. In that case, the possibility of duplicates cannot be excluded. That is why we talk about secondary matching keys or key variables.

5.2 Applicability

The assumption underlying this method is that the matching keys used in both files are of good quality. However, this is not always the case: the quality of the matching keys may be unknown, or it may not have been sufficiently investigated. Still, this form of matching is used very frequently. The method is simple and not too difficult to perform. It also does not require a lot of knowledge.

The use of this matching method is very broad; it can be used in those cases that have high-quality unique primary matching keys.

If the quality is not as good as hoped, but if there is a full list of primary keys with information about the associated units, then another action may still be possible. Suppose that you are using the BSN as the primary key, and you also have a complete list with BSNs with at least some information about the people concerned. If you come across a BSN that does not seem correct, then you could look 'close by' this number in the list. The idea here is that a mistake was made when copying the number, for example, two digits were interchanged, or a 5 was replaced by a 6 (or vice versa) or a 7 by a 1 (or vice versa), etc. If, for example, you search for all BSNs with a Levenshtein

distance of 1 or 2 (see Section 3.2) from the given BSN, and also compare the associated personal attributes with the data in the file or register concerned, you could potentially find the correct BSN with the associated personal attributes. This is, in fact, a method that belongs in Chapters 6 and 7.

5.3 Detailed description

In view of the simplicity of this method, there is not much more to be said about it. The basic principle is that a match is made if and only if a record from one matching file has exactly the same key values as that of another record from the second matching file. This means that there can be 1:1, 1:n or n:1 matches. If the quality of the primary matching key is insufficient or unknown, there is a danger that matches will be made that are not true matches (mismatches), and that should have been made but were not (missed matches).

5.4 Examples

We offer the following examples of matching situations where matches are made based on a primary key:

- The matching of enterprises from two statistics, which are both based on the General Business Register (ABR). In both files, the unit – the enterprise – is identified by an eight-digit business identification number (a BEID). The BEID is the primary key on which matching takes place. If the BEIDs in both files are the same, then a match is made; if the BEIDs are not the same, then the files are not matched. For example, no account is taken of the fact that, during the processing procedure for the individual statistics, errors could have crept into the BEIDs. This check is also often difficult because, in many cases, there are no more secondary keys present, such as names and addresses;
- A variation on the first example is that data from the Tax Administration is matched with the BEIDs from the ABR. The one file from the ABR contains the BEID as the primary key. The file from the Tax Administration contains the Tax Group (FE) as the primary key. To match the two files, a 'relationship or matching table' is present, which indicates which FEs are associated with which BEIDs. In the same way as in the first example, the two files are matched, but with an extra step. There is a higher risk of incorrect matches here, because errors could have crept not only into the FEs or the BEIDs, but also into the registration of the relationship between the FEs and the BEIDs;
- For personal details, the citizens' service number (BSN) is often present as a primary key in the file. In such cases, a simple match can be made based on the BSN in the two files. An example is the matching of salary and employment data from the Tax Administration and from the policy administration (of the UWV, the Dutch social security benefits administration);
- Matching based on a foreign key, for example, the SBI coding or size class coding in a record of a BEID. Indicators or averages from a file based on the SBI or size class, but then as a primary key, can be matched with the record from the file with BEIDs. This occurs frequently in editing or imputation.

It should be noted that an external primary key for matching activities at Statistics Netherlands can be replaced by a key that is only meaningful – and therefore can only to be used – at Statistics

Netherlands. This process involves assigning a RINs (and removing the primary key). The purpose of this is data security, to prevent a situation where various other types of information can be matched with a file based on an external primary key, such as the BSN.

5.5 Quality indicators

The quality of this type of matches is totally dependent on the quality of the primary key. Oftentimes, there is a tendency to presume too easily that the quality is sufficient. It is also possible that only the primary key is present in the file and no secondary keys are available. That is the case, for example, if files had the primary keys been replaced by RINs.

The quality can be checked here by taking a small sample from the unmatched and matched pairs of records and then examining these manually using the other variables in the record, while also looking at, for example, complex units or deviations thereof (compare: outliers). In this context, Type I and Type II errors (and estimations thereof) can be used as quality measures.

5.6 Variation

A variation of this method is when the matching does not take place directly based on the primary key, but where there is a relationship or matching file. In that case, it is still possible to match different units.

6. Matching based on secondary keys, without matching weights

6.1 Short description

Matching based on a secondary key utilises one or more variables that possibly identify the record. This identification is not necessarily unambiguous, unlike when using a unique primary key. The problem is that a unit that is unique based on a set of secondary keys does not necessarily have to be unique in the population. In the Dutch population, there are many people who have 'Janssen' as their surname, or who have the profession of 'civil servant'. Even combinations of several such variables can still produce duplicates: there are multiple civil servants with the surname of Janssen. In contrast, someone with the surname of 'Wladimirow' whose is a lawyer by profession could be unique in the Netherlands (at a given point in time). The more *direct or indirect identifying variables*[15] available (such as initials, first name, surname, business name, gender, date of birth, age (at a certain point in time), address, profession, etc.), the greater the chance that unique people will be designated in a file. They do not all have to be unique, but some of the people represented in a file could be unique. The more such variables are present in a file, the more uniques (in the population) you will find and not only in the file. If the scores are reliable, then these are probably also actually population uniques.

Additionally, observation errors and other deviations can also occur in the values taken by these indirect identifiers. In that respect, they differ from the variables that are used as primary key variables. Furthermore, 'values-with-deviations/errors' based on secondary keys are usually still usable for matching. Errors based on scores of primary are usually not useable (for example BSN numbers with typing errors).

6.2 Applicability

A condition for using this method is that common indirect identifying variables[16] are present in both matching files, based on which the match can be performed. We also allow the situation that two similar variables have a different domain, for example, with another category division (for example, age in five-year classes in one file, and in ten-year classes in the other). We also accept that observation errors can occur in the scores of these variables.

In practice, the decision to work with a matching method that does not use matching weights is often connected with performance. If the files to be matched are large, these methods generally

[15] These names are derived from statistical diclosure control. See Willenborg and De Waal (2000). For that matter, we are not talking about primary keys such as a social security number or BEID here. We do mean variables such as surname, initials, first name, address, etc. These are indeed direct identifiers, but they do not always designate unique units. After all, a surname such as 'Janssen' occurs frequently, such as the place of residence 'Amsterdam', or the address 'Dorpstraat'. When combined, they are much more powerful, and can indicate unique units such as people. A score or BSN always indicates a unique person.

[16] Their having common variables is not enough. They could be, for example, variables that express an opinion or viewpoint. For example, answers to question such as: Which party did you vote for in the last election? Do you feel safe on the street at night? The answers to these types of questions are generally not very reliable.

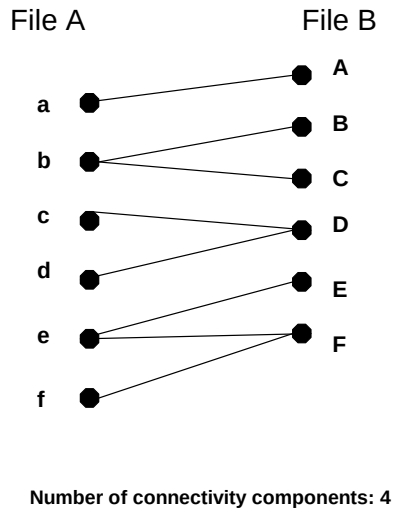
work faster than those with matching weights. However, one should expect that the quality of the matches made – in terms of Type I and Type II errors – is generally lower.

6.3 Detailed description

An MC graph indicates which records are matching candidates based on the matching criterion used. An MC graph is compiled after applying the matching criterion to every possible pair of records.

An MC graph is formally a bipartite graph and is defined as follows for a matching problem where there are two files A and B, and a matching criterion K used on a matching key S. For an example of an MC graph, see Figure 6.1.

Figure 6.1 Example of MC-graph without matching weights



We take files A and B to be sets of records. $G = (V, E)$ is the MC graph for this matching problem. The node set V is given by $V = A \cup B$ and the edge set E consists of the pairs $\{a, b\}$ where $a \in A, b \in B$ which furthermore satisfy the matching criterion K.

An MC graph is depicted in Figure 6.2. The edges indicate the matching candidates. The match $\{d, h\}$ is the only one that can be made unambiguously, separate from the matching criterion used. Depending on the matching criterion, more matches can be made. If this concerns a 1:1-matching, two additional matches are possible:

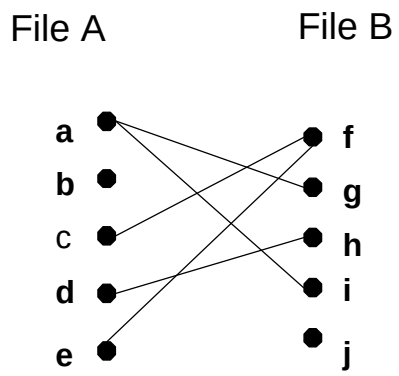
1. $\{a, g\}$ or $\{a, i\}$ (one of the two)
2. {c,f} or {e,f} (one of the two).

The choices in 1. and 2. can be made independently of one another.

In the case of an MC graph without weights, the candidate matches all count the same in terms of weight. An important example of a matching criterion that results in an MC graph without matching weights is that of equivalent scores on the matching key. The matching method based on this criterion is also called *exact matching*. It should be noted that this only refers to the fact that the matching criterion used requires exactly equivalent scores on the variables in the matching key for the associated records to be considered as matching candidates. It has nothing to do with 'accuracy', or the matches being 'error-free'. The reason for this is that, in practice, errors, deviations or irregularities occur in the files to be matched, and more particularly on the matching key. These errors in the data lead to a situation where matching candidates are found that do not relate to the same units, or a situation where matches are missed.

If we broaden the scope of the matching criterion for exact matching, we can search for matches of

Figure 6.2 MC-graph without matching weights



units with a small number of deviations in the scores on the matching key used. Suppose that the matching key consists of the variables (secondary keys) $s_1,...,s_k$. Suppose that a is a record from file A and b is a record from file B. We show the scores of a and b respectively as $(s_1^a,...s_k^a)$ and $(s_1^b,...s_k^b)$. The records a and b are matching candidates if $(s_1^a,...s_k^a)=(s_1^b,...,s_k^b)$. Instead of this, we could also allow deviations to be present, but a limited number, say a maximum of p. If, for example, we use the Hamming distance d_H (see also Chapter 3), we could consider records a and b as matching candidates if $d_H(a,b)\leq p$, and not as matching candidates if $d_H(a,b)>p$. Something similar would also apply with a different metric.

The approach described above is rather black and white: two records are either matching candidates or they are not. We could also introduce a zone of doubt. In the case of the Hamming distance, we

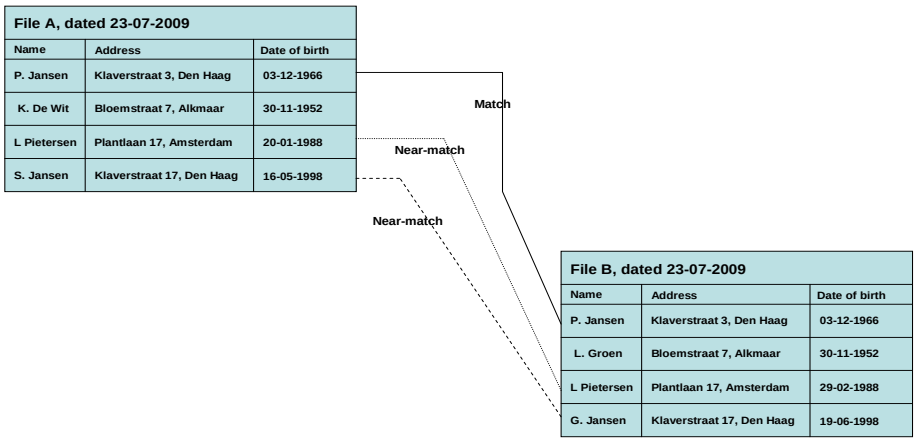
can use two parameters (natural numbers) p, q where $p < q$. We then use the decision table presented in Table 6.1

Table 6.1 Matching candidates, doubtful cases, not matching candidates

Condition	Interpretation
$d_H(a,b) \leq p$	a and b are considered matching candidates
$p < d_H(a,b) \leq q$	a and b are doubtful matching candidates, and should be inspected by a specialist on the subject who should determine whether or not a and b are matching candidates
$d_H(a,b) > q$	a and b are not considered matching candidates

The parameters p and q can be chosen so as to control how many matching candidates have to be inspected. The choices may be guided by the available capacity of specialists who can assess the doubtful cases. The parameters p and q are examples of cut-off values.

Figure 6.3 Matching using a Hamming metric



6.4 Example

Suppose we have two files, A and B, and want to match these, based on the common matching variables of name, address and date of birth (all secondary keys). First of all, we only want to match the first records from both files (P. Jansen). There is a match when there are equal scores on all the matching variables. Then, we relax the requirement of equal scores on the matching variables of name, address and date of birth. For the variable of date of birth, matching based only on the year of birth is sufficient. This produces an extra match for the name L. Pietersen. Finally, we relax the requirement even further. There is a match if the surname, the year of birth and the address are the same. This produces an additional match, the match between S. Jansen (file A) and G. Jansen (file B). This situation is shown in Figure 6.3. This is an example where we could also

argue that a metric was used: a Hamming metric (see Section 7.3.1.1). However, this metric leads to the weights 0 (not a matching candidate) or 1 (matching candidate).

There are also comparable examples in economic statistics, such as if matching is performed based on matching variables like 'enterprise name', 'address' and 'telephone number'. It should be clear that this is no easy task, because enterprise names can be recorded in many different ways. For example, the enterprise may be recorded under the name of the formal legal entity (such as Verkoop Vanalles BV), or under a shortened version of the name (such as Vanalles) or under the name of the owner (such as G. Jansen).

6.5 Quality indicators

The number of mismatches or missed matches can also be used as quality measures in this context. A few issues play a role here, and these correspond with the crucial steps in a matching process:

1. Finding matching candidates. The following issues play a role:
    a. The matching criterion used (for example, using the Hamming distance) to consider records as matching candidates or not.
    b. If you use a metric, etc., the question is to what extent this adequately takes into account the underlying error process. (See also Section 7.3.) The choice of cut-off values also influences which records are considered as matching candidates.
    c. Any blocking variables used (for example, in large files); by partitioning the matching files and intentionally limiting the searching space (because of performance), you may miss candidate matches, and ultimately also matches.
2. Selecting the final matches from the matching candidates. A criterion is used here too. The question is the extent to which this leads to a correct choice.

The quality of a matching method used can be assessed based on the inspection of matches of test files. It is a labour intensive job to carry out. You must examine not only the matching candidates and the matches ultimately selected, but also any missed matches under various parameter settings.

6.6 Variant: use of a distance function

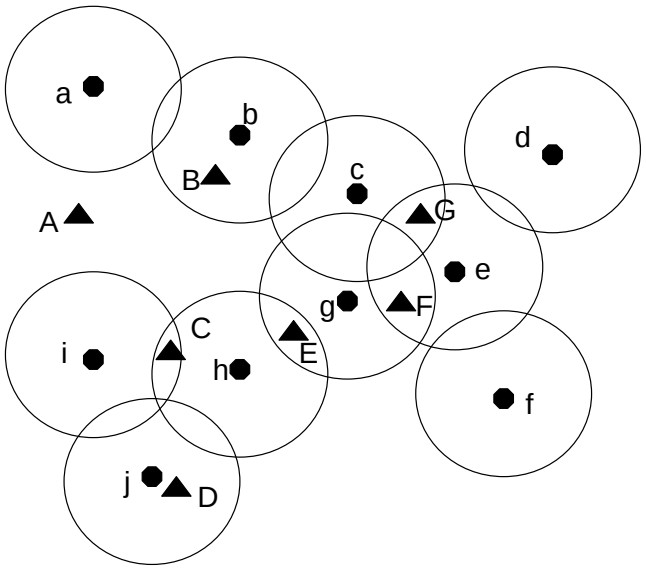
For matching based on a primary key, it is required that records have the exact same score on the matching key used. We can also use this matching criterion using secondary keys, but this method is less attractive here. We can relax this requirement and consider two records as candidate matches if the scores for at least k (parameter to be established) of the maximum n (length of the matching key = number of matching variables in the matching key) are the same. In fact, a metric is used here, the so-called Hamming distance or Hamming metric.

We now give some comments on the approach taken in the present chapter. If we denote a Hamming-metric by d_H [17] and we interpret the scores on the matching key as vectors of length n, then the matching criterion used here is in fact:

[17] In this notation, the length of the matching key, n, is intentionally not included in order to keep the notation simple.

For $\alpha \in A, \beta \in B$, α and β are matching candidates if and only if $d_H(\alpha, \beta) \leq k$.

Figure 6.4 Records from two different files represented as points and environments of records from one of the files



Point: record from file A; Triangle: record from file B; Circle: neighbourhood of a record from file A. Preferably choice the file with the "hard" matching data as the file from which the neighbourhood is determined to see if matching candidates are available. The choice of the radius of the circles is also a point of interest: the more matching candidates the bigger, but also the chance of more mismatches.

We can formulate this in such a way that all β from B that are inside a circle with radius k (using d_H as a metric) around α provide all matching candidates for α. See also Figure 6.4.

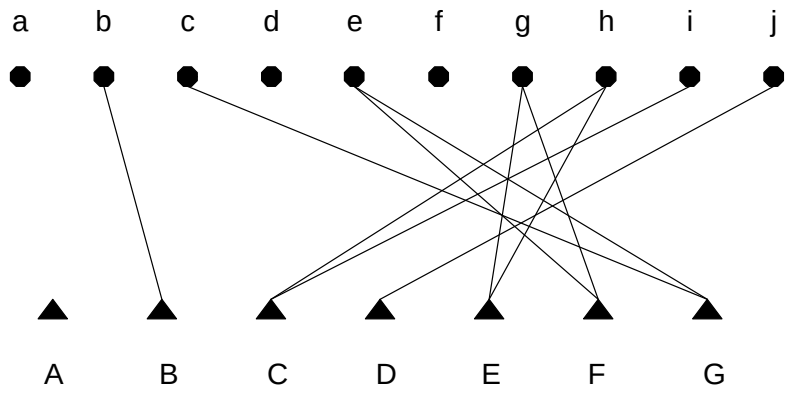
For each α in A, we can ascertain this in B. (Or vice versa, for each β in B, we can figure out which α in A are inside a circle with radius k around β. That produces the same result). Note that, here, we only use whether a record is present in a circle around a 'point', not at what exact distance it is from that point. We could, in fact, use this distance as a matching weight: the smaller the distance the higher this weight. In Chapter 7 this approach is described.

If we look at the above section critically, we can conclude that the selection of a Hamming distance is not essential for the approach taken; we could just as well have chosen another metric to arrive at a similar matching criterion. Therefore, based on a metric d, it is possible to formulate a matching criterion:

For $\alpha \in A, \beta \in B$, α and β are matching candidates if and only if $d(\alpha, \beta) \leq k$.

Once again, we can formulate this in such a way that all β from B that are inside a circle with radius k around α (measured using d), provides all matching candidates for α. For each α in A, we can ascertain this in B. (Or vice versa, for each β in B, we can figure out which α in A are inside a circle with radius k around β). All of this produces an MC graph, where records from A and B are represented, and those which are matchable are connected by an edge. See Figure 6.5.

Figure 6.5 MC-graph, representing the situation in figure 6.4.



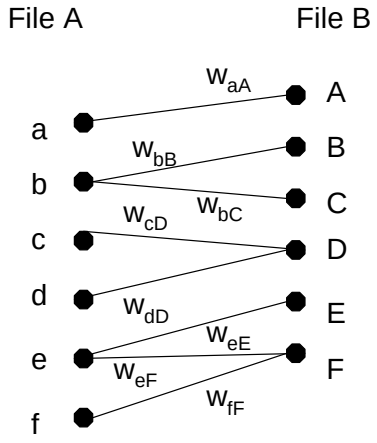
Suppose a 1:1 match. Matching candidate bB is a certain match. With cG, eF,eG, gF there are two alternatives: {cG,eF} or {eG,gF}. Without extra information a choice cannot be made.

7. Matching based on secondary keys, with matching weights

7.1 Short description

Various matching techniques make use of matching weights, which can be used to differentiate between the potential matches represented in an MC graph for a matching problem. See Figure 7.1. There are a variety of reasons to work with matching weights: you may want to express that not all of the variables are equally reliable, that is, that they do not have reliable scores. Alternatively, you may want to indicate that the units corresponding with records that are matching candidates demonstrate a certain degree of similarity or dissimilarity. Or you may want to demonstrate that thery are a certain distance apart, as measured by a certain metric. Or you want to use a probability to show that two units are the same. Then a probability model is needed to quantify differences in scores on the matching key.

Figure 7.1 MC-graph with matching weights



The higher/lower the matching weights, the more/less the connected units are close to each other or similar. The relationship between the matching weights and the degree of similarity, must be determined per case.

7.2 Applicability

The matching method without matching weights could be characterised as black and white: two records are either matching candidates or they are not. There is no room for any nuance. However, there are situations where it is desirable to add this nuance. Consider variables such as first name, surname, street name, city/town name, etc. Here, you may want to express the extent to which two surnames differ from one another. The difference between ‘Jansen’ and ‘Janssen’ is smaller than the difference between ‘Jansen’ and ‘Cuypers’. This concerns purely the spelling of the names: the letters that are present and their order of occurrence. However, this can be quantified using a metric (see Section 7.3.1.1). In other examples, it is not so much about the extent to which strings differ from one other, but the extent to which the meanings of the strings are different. This is, for

example, the case for professions. The words (concepts) ‘teacher’ and ‘instructor’ are very different from one another as strings, but in terms of meaning, they are very close, and could even be considered the samen being synonyms. This concerns a different distance concept than the one discussed above. The distance concept here relates to the meaning or semantics associated with strings understood as words or concepts. A similar difference is obtained if we do not look at how the strings are written, but at how they are pronounced. In Dutch, ‘Taylor’ and ‘Teler’ are pronounced almost the same; they are close to one another phonetically.

In both cases, we do not measure the distance ’stringwise’, but semantically or phonetically. Let s, t be two strings and $f : S \to T$ is a mapping of the set S of strings to a space T of meanings, or of pronunciations. Let d be a metric on S, and D a metric on T. Then $d(s,t)$ measures the distance between the strings s and t and $D(f(s), f(t))$ the distance between the meaning of s and t, or their pronunciation.

A metric is an example of a function that can be used to calculate matching weights. These matching weights can be used to express the strength of a candidate match. We should add that, in practice, it is necessary to work with cut-off values: matches that are too weak in terms of the associated matching weight are not considered to be matching candidates. The trick is to properly establish these cut-off values: not such that too many irrelevant matches are made, but that the correct matches are not missed. In practice, this requires experimentation with various settings of the cut-off values.

Other possibilities to arrive at matching weights without using metrics are discussed in Section 7.3.1. All the considerations to use matching weights must be dictated by the processes or mechanisms that have given rise to – or could give rise to – differences in the data. This could be writing mistakes (‘Jansen’ instead of ‘Janssen’), or the use of alternative designations where possible (for addresses: ‘Dorpsstr.’ instead of ‘Dorpsstraat’); for professions: ‘instructor’, ‘teacher’, ‘lecturer’, ‘tutor’, all indicate similar functions in education. It is therefore important to have thorough knowledge of the way in which the files to be matched are compiled. In addition, it is possible that not exactly the same matching variables will be used in the two files, or that the scores do not relate to the same moment in time. As a result, the attributes of an entity (individual, business, etc.) could have changed.

7.3 Detailed description

7.3.1 Calculating matching weights

There are different ways to determine matching weights that can be used in a matching problem. We will discuss several of these ways here. The list is not exhaustive, but it does provide several important examples. These matching weights are used for matching if the information about the ‘matching candidacy’ of two records is not represented in ‘either/or’ form (matching candidate? ‘yes’ or ‘no’), but with more nuance. The extent to which two records match can be expressed in a matching weight.

In the discussion in the sections below, we look at two files, A and B, that contain records, for which there are common matching variables $v_1,...,v_n$ that together form the matching key, based on which the records in the two files are matched.

7.3.1.1 Based on metrics or generalised metrics

Earlier in this document (Chapter 3), we discussed metrics and generalised metrics in a general sense. In that chapter, what was mainly important was which attributes a metric posseses, as well as pseudometrics and other varieties. For matching, it is important to find suitable metrics for each of the variables in a primary or secondary matching key, or rather for each type of variables. The following variables may occur in a secondary matching key: names (first names, surnames, enterprise names, street names, city names, etc.), time indications (dates of birth, ages at a certain reference time), gender, marital status, professions, etc. Finding suitable metrics for variables to be used as secondary matching keys can be seen as a separate subfield in matching.

A general problem is to find new structures from given ones. In particular, this applies to sets with metrics. For a given set X with metric d – the pair (X,d) is also called a metric space – for each $Y \subseteq X$ the metric $d|_{Y\times Y}$ (the restriction of d to $Y \times Y$), produces a new metric space $(Y, d|_{Y\times Y})$.

Another example is to attach a metric to a product set, if all component sets have a metric. Suppose that we have several variables with a metric defined on the associated domains. Let $X = D_1 \times ... \times D_n$, where D_i is the domain of matching variable v_i with metric d_i, then a metric d_w on X, for example, can be defined as

$$d_w(x,y) = d_w((x_1,...,x_n),(y_1,...,y_n)) = \sum_{i=1}^{n} a_i d_i(x_i,y_i),$$

for some constants $a_i > 0$ for $i = 1,...,n$. In practice, the weights a_i can be used to coordinate the relative importance of the metrics-per-variable mutually. The variant described in Section 6.6 also uses a metric, while this may not so obvious at first glance.

Another way of defining a metric on X is as follows:

$$d_{\max}(x,y) = d_{\max}((x_1,...,x_n),(y_1,...,y_n)) = \max_{i=1,...,n} d_i(x_i,y_i).$$

In this case, all of the submetrics have the same weight in terms of strength. Here, we can also differentiate between the strength of the submetrics using weights $a_i > 0$, $i = 1,...,n$ to define:

$$d_{\max,w}(x,y) = d_{\max,w}((x_1,...,x_n),(y_1,...,y_n)) = \max_{i=1,...,n} a_i d_i(x_i,y_i)$$

A metric that can be defined for every variable is the 'black-white metric' d_{01}, defined as

$d_{01}(u,v) = 0$, if $u = v$,

$d_{01}(u,v) = 1$, if $u \neq v$.

Based on this black-white metric, the Hamming distance can be defined per variable as

$$d_H(x,y) = d((x_1,...,x_n),(y_1,...,y_n)) = \sum_{i=1}^{n} d_{01}(x_i,y_i),$$

That d_{01} and d_H can be used 'universally'– regardless of the type of variable(s) – is a strength as well as a weakness. In practice, there is actually a need for metrics that are tailored towards specific variables, or types of variables. These metrics also add more nuance, in the sense that they quantify some differences as more or less than others.

We provide several examples of such metrics here and in Appendix B. The goal is mainly to illustrate a number of possibilities.

An important case concerns variables in which alphanumeric strings constitute the scores, such as in name (first name, surname, street name, enterprise name, etc.). If we interpret such a name as a string of symbols, we could compare two strings σ and τ using a Levenshtein metric d_L, introduced in Section 3.2. For more information about this metric and for applications in several fields of study such as biology, linguistics and bioinformatics, see, for example, Sankoff and Kruskal (1983, pp. 18, 19).

Example: Consider, $d_L(Janssen, Jansen) = 1$ (remove 's' from the first string) and $d_L(Hendricks, Hendrikx) = 2$ (remove 'c' from the first string and change the 's' to 'x').

This definition of distance is based on the written text (name). However, it is sometimes better to take account of the pronunciation of the text (name). In practice, there may be spelling variations such as 'Janse', 'Jansse', 'Jansen', 'Janssen', 'Janszen' and 'Janzen', and 'Hendriks', 'Hendricks', 'Hendriksz', 'Hendrix', 'Hendrikx', and 'Hendrickx'. Here, you may actually want to use a function that represents these names phonetically. In that case, the spelling variations of 'Jansen' and 'Hendriks' respectively would be mapped onto the same image. For a – slightly more technical – further discussion, see Appendix B. Several other metrics for strings are discussed there.

Trigrams are also used to compare strings with each other. Trigrams can be used to relate strings with – a limited number of – spelling deviations. The concept of trigrams is illustrated in the example below.

Example: Take the names '_Hendriksz_' and '_Heinrichs_' (at the start and end of each string, we add a space ('_'); we will look at these extended strings). The trigrams for the first string are (we write everything in lower case letters): (_he, hen, end, ndr, dri, rik, iks, ksz, sz_) and for the second string (_he, hei, ein, inr, nri, ric, ich, chs, hs_) .

Note that we have left the trigrams above in order, in rows. However, we can also examine the associated sets, in this case {_he, hen, end, ndr, dri, rik, iks, ksz, sz_} for the first string and {_he, hei, ein, inr, nri, ric, ich, chs, hs_} for the second string. Based on these sets of trigrams, we can easily derive a metric by counting the number of trigrams that occur uniquely in the two strings. If we have two of the same sets of trigrams, say S and T for strings σ and τ, respectively, then

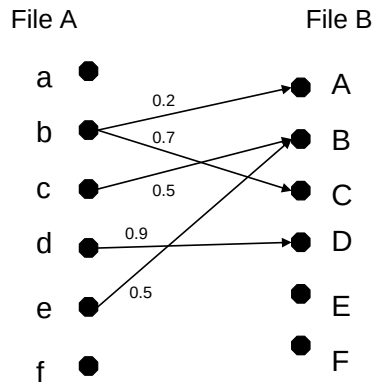
$$d_{tri}(\sigma,\tau) = |(S \cup T) \setminus (S \cap T)| = |S \setminus T| + |T \setminus S|$$

is a metric. We could also define another metric that takes account of the number of corresponding characters in the set of unique trigrams of the two strings. However, we will not discuss this further, because that would be getting too far off the subject.

For variables where the domain has a natural (partial) order, we can use the associated tree or directed tree to determine the distance between each pair of points in the domain. This is equal to the length of the shortest path in the tree that connects the two points. Here, each edge has a length of 1.

There are even more metrics (and related functions) that can be derived for matching variables. This can be seen as a specialism in the field of matching. We will not discuss this further, because it leads us too far from the core of this document.

Figure 7.2 MC-digraph with matching weights, in this case with probabilities



It looks like unit b in file A has developed into two units in file B (A and C). B in file B is an unit which could have been developed out of two units in file A (c and e). This case could relate to two files with different timestamps.
Note that the probabilities (from b, or to B) do not add up to 1. This could mean that there is a chance that b is not developed to either A or C. With B the chance that this unit is developed out of c or e is estimated as 1. Alternatives for these two options are apparently excluded.

7.3.1.2 Based on probability

Matching weights can also be based on probability models. Stochastic methods can enter into matching for different reasons. See Figure 7.2. We offer the following reasons:

1. Errors can occur in the secondary matching keys. The errors can be present for various reasons. An answer to a question in a survey could have been understood incorrectly and therefore answered incorrectly by the respondent in question; a given answer could have been incorrectly processed, for example, keyed in wrongly; errors could have been made in the coding of answers, etc. This type of error is often referred to as a non-sampling error. The first step would be to identify and model all major sources of errors using probability models. These models can then be used to calculate the probabilities that two scores match based on corresponding secondary keys from two matching files.
2. The reference times of the two matching files differ to such an extent that the effects of the dynamics of the population are noticeable on the units contained therein: values of certain scores could have been changed for some units. A person could have turned a year older; an enterpise could have merged, split or gone bankrupt; a person could have changed jobs; an unemployed person could have found a job, etc. Therefore, if the reference times differ significantly from one another, it is not self-evident that the units and/or their scores on secondary key variables would have remained unchanged.
3. Some comparable matching variables are not defined exactly the same way in the two files. The associated question can be different, or the position in the questionnaire could have been changed, or the value range of comparable variables may differ slightly. In that case,

it may sometimes be unclear which scores correspond with one another. Suppose {20,21} is an age class in one matching file and 11 - 20 and 21 - 30 are age classes in the other file. The 20 and 21-year-olds are in the same age group in the first matching file, but they are in two different age categories in the second. We can also estimate which part of the people in the category (20,21) in the first file will end up in the age category 11-20 and which part will be in the age category 21-30 in the second file: $\frac{n_{20}}{n_{20}+n_{21}}$, and $\frac{n_{21}}{n_{20}+n_{21}}$ respectively, where n_{20} is the number of 20-year-olds on the measurement date and n_{21} the number of 21-year-olds at that point in time.

In practice, combinations of these causes of differences often occur. Files can have different reference times, there may be processing errors in the data, and the units may not be exactly comparable. Section 7.4 presents an example of a situation as in point 3 above, and an example with a combination of points 2 and 3 above (variables with deviating value ranges and different reference times).

7.3.1.3 Weights for the quality of matching variables

In practice, based on the quality of the scores, we will want to differentiate between the different matching variables in the matching key. Some variables will have more reliable scores than others, and we will want to take this effect into account when determining the overall matching weight.

We consider this 'quality weight' as a subjective weight that the person performing the matching establishes based on his/her knowledge and experience with the different variables in the matching key. It is possible that experiments must first take place before a good choice can be made about these weights. These weights only have meaning in terms of the relationships between them, not in an absolute sense.

In the discussion about multivariate metrics in Section 7.3.1.1, weights are introduced that can be established by the user.[18] Users can express the relative importance of a variable for the multivariate distance function. In this way, they can influence the effect of a certain variable in the total. If the variable has been reliably measured, then a relatively high weight is needed. If it is a variable with relatively more errors than the other variables in the matching key, then this variable should be given a lower weight.

For that matter, it is also possible to express the difference in the quality of matching variables in a different way, for example, when matching, by going through the scores in the order of the quality of the matching variables (from high to low), and then accepting certain deviations in the scores with increasing tolerance.

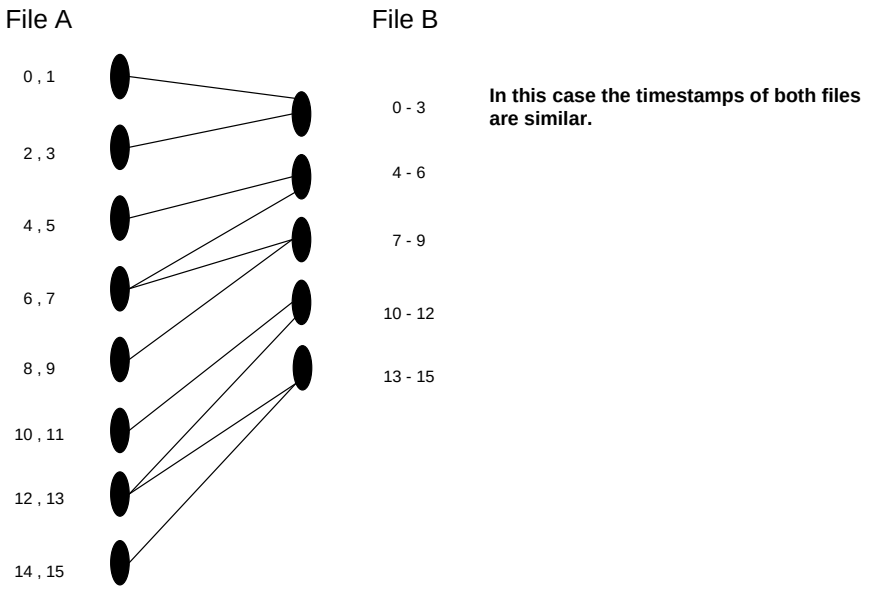
7.3.2 MC graph with matching weights

Once we have selected a method to determine matching weights, we can start calculating an MC graph with matching weights. We may have to use a cut-off value so that we do not have to include

[18] For the mathematics, it does not matter which combination of weights is selected. As long as all the weights are >0, the result is a multivariate metric.

candidate matches of two records with a matching weight that is too low (they will not become edges in the MC graph).

Figure 7.3 Two age variables and their relationship. One variable specifies age in two-year classes (file A) and the other specifies age in three-year classes (file B)



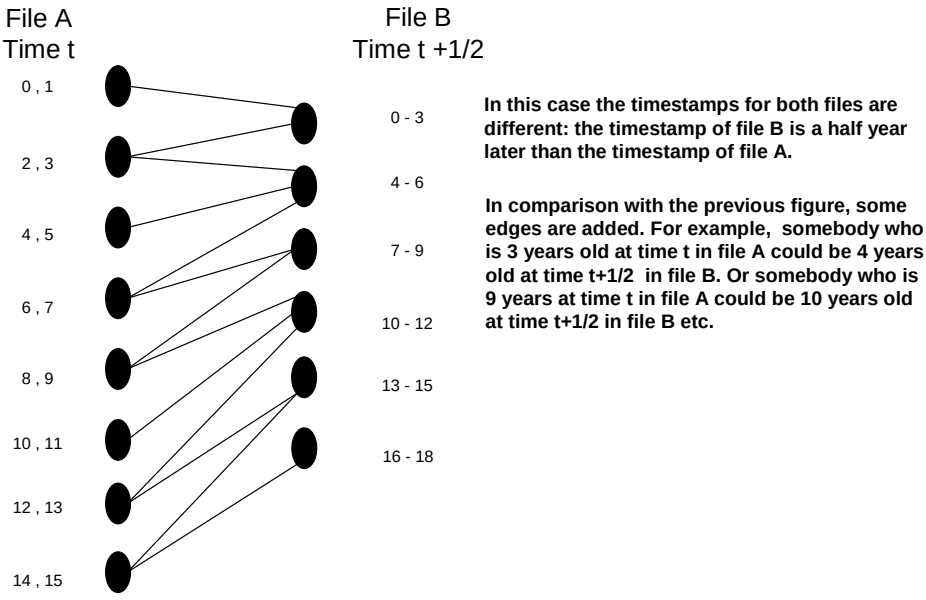
7.4 Example

Here we discuss an example of a situation in which two matching variables are similar but not exactly the same. Specifically, we are talking about two age variables.[19] One of the age variables, which specifies age in two-year classes, occurs in matching file A, and the other, which represents age in three-year classes, is found in matching file B. Depending on the reference times for each file (the time to which the data relates), we can make a connection between the age categories. Figure 7.3 shows a graph that relates the age categories from the two files if the reference times are the same.

In practice, the reference times of two matching files do not have to be exactly the same. Indeed, it is more likely that they will not be the same. Moreover, in practice, the data tends to relate more to an interval than to a specific point in time (for example, because the individuals concerned will not all be interviewed at the same point in time). In Figure 7.4, a connection is made between the age categories if the reference times for the two files differ by a half a year. In this case, some people may have turned a year older in the interim.

[19] We assume that, in both cases, the age represents: the age at the reference time of the survey concerned. For example, the answer to the question: 'How old are you now?'

Figure 7.4 Two age variables and their relationship. One variable specifies age in two-year classes (file A) and the other specifies age in three-year classes (file B)



7.5 Quality indicators

The quality indicators with regard to mismatches and missed matches - referred to in Section 6.5 and elsewhere - apply here as well. These are influenced by the way that the weights are calculated, the use of cut-off values and the use of blocking variables to stratify large files. See the discussion in Section 6.5.

7.6 Variants

The sections in this chapter place the emphasis on the basic variant for matching with matching weights, where the matches are 1:1. As stated earlier, there are also situations in which 1:n, m:1 and even n:m matches are possible. This is the case for composite units such as businesses which, over time, can split or merge into other units. Formally, this means that the conditions under which matches are possible must be adapted. Also they do not relate to the same units, but to combinations of units that produce comparable entities. See also Section 9.3.

In the discussion we have so far assumed that all the scores on secondary keys are present. In practice, however, this is not always necessarily the case, and scores can also be missing. Calculating matching weights is more difficult in this situation, because the missing values cannot just be omitted: they must be replaced by stochastic variables, with a known assumed distribution. In such cases, the unknown parameter values must be estimated using, for example, the EM algorithm. For information about the EM algorithm, see Wikipedia (http://en.wikipedia.org/wiki/EM_algorithm) or the references stated there.

8. Matching software and IT considerations

8.1 Matching software[20]

Data matching is virtually impossible without the use of a specialized software package. An automated approach is required due not only to the size of the files and the sets of matching candidates, but also the cleaning and parsing of the data in the pre-processing phase and the calculation of weights and the determination of the subsets of matched and non-matched records.

There are different software packages on the market that can be used for matching. Most of these are commercial software, which are increasingly embedded as a component in packages for business intelligence an data mining. Trillium is an example of such a package. In that case, the methods used are generally a 'black box' for the user. A limited number of packages are open source, such as Febrl and the Link King. Nearly all of the packages are geared towards matching persons. There seem to be no packages that focus specifically on matching businesses.

All of the matching packages have their own advantages and disadvantages. Several options are discussed briefly here.[21]

Standard software

- First of all, *standard software packages* can be used for matching, such as MS Access, MS Excel, SQL, SPSS, Clementine, SAS[22] and Manipula. These packages are not made specifically for matching and are therefore only appropriate for simple forms of matching, such as joining. If you want to use more complex methods with, for example, matching weights, then these packages are unsuitable. These packages also do not support activities in the pre-processing phase, such as parsing, blocking and specific comparison components, such as Soundex. Oracle and Microsoft plan to bring more advanced matching software on the market as part of their database management systems.

Freeware

- **Febrl** *(Freely Extensible Biomedical Record Linkage; www.sourceforge.net)*. This package is available as freeware and open source (Pyton). The package, which was specifically developed for matching data, can be adapted as desired according to your own specific wishes and methods.
    - *Package:* For the pre-processing phase, it contains options for such things as editing, cleansing and parsing data. Blocking variables can also be used, and there are possibilities for deduplication. In this phase there are options to create various summaries of the data. Test sets for test runs can be generated. The package has a GUI and is relatively easy to use. Febrl therefore covers the entire matching process, and not only the matching phase.

[20] Software development is an ongoing process. This means that in the meantime new software could have come available and that existing software could have come obsolete.

[21] Packages were not examined in detail. In most cases, we only visited the website of the supplier in question. The descriptions given here are therefore general. More research is desirable for a subsequent version of this report, so that the various packages can be weighed up against each other.

[22] SAS has somewhat more extensive opportunities for matching. See the Dataflux package.

- *Method:* Febrl allows various matching methods to be used (and adapted by the user), including in particular methods with weights (probabilities based on Fellegi and Sunter; cut-off values; based on, for example, Hamming, Soundex and Q-grams), resulting in matches, non-matches and doubtful cases. Different options are available to calculate the weights.

- **The Link King** *(Record linkage and consolidation software; www.the-link-king.com)*. Just as Febrl, this is freeware. It is written in SAS.
  - *Package:* The system offers the possibility of deduplication and blocking. In the comparison of key variables, there are options for, for example, Jaro-Winkler string comparisons, Soundex and others, metrics (for postal codes and strings, for example), conversions of names, and weights of names (the name Smith has a lower weight than Walofski in the Netherlands). It has a GUI that includes a specific component to examine doubtful cases. The package can randomly generate a set for a test run.
  - *Method:* The package supports both methods without weights and joining, as well as methods with weights (probabilities). The GUI offers support in the selection of the correct methods (based on Artificial Intelligence). Major disadvantages are that The Link King requires an SAS licence and only processes person records.
- **Link Plus** *(www.cdc.gov/cancer/npcr)*. This is a stand-alone matching programme and is also freeware (however, the code is not available).
  - *Package:* The package matches two files with one another. With regard to the pre-processing phase, it only offers the option of deduplication. It also supports the manual processing of doubtful cases. Blocking is possible. There are comparison options based on names.
  - *Method:* This package uses methods with weights. A drawback is that, just as much of the other matching software, it mainly focuses on files with people.

Commercial packages

- **Trillium** *(from Harte-Hanks; Trilliumsoftware.com)*. Trillium covers the entire matching process. It should be seen as a set of functions around a database management system.
  - *Package:* Trillium offers many options for the pre-processing phase (to improve the data quality; TS-Quality module) such as a parser to cleanse, deduplicate and standardise the data, and a geo-coding system to check address information (ETL tool). This package also has a module that looks for inconsistencies in and across files (TS Discovery). The desired manipulations can be compiled relatively easily using a GUI (compare with Clementine). Trillium can match more than two files simultaneously. It is also possible to use a composite key. The result is recorded in three classifications: ‘pass’, ‘fail’ or ‘query’ (the doubtful cases). It can be used both online and in the batch. In addition, it is possible to keep track of all actions and changes in an audit trail.

- *Method*: The product has various matching options based on matching with weights (Trillium parallel matcher; not Fellegi and Sunter). Trillium has been around for a while (since 1989) and is one of the commercial leaders in this area (see Gartner, 200723).

- **GRLS 3** *(Generalized Record Linkage System;* http://www.statcan.gc.ca*)* is a commercial matching package from Statistics Canada. It is written in C and works primarily with Oracle as the DBMS. It was specifically set up for cases where no unique primary key is present, and is suitable for files with both people and businesses.
  - *Package:* GRLS has two steps 1) determine the matching candidates based on criteria to be determined by the user (decision rules) and 2) determine whether there is a match with the following as the results: 'definite', 'possible' or 'excluded'. Deduplication is also possible. In comparisons between key variables, it is possible to use, for example, Soundex. Test files can be made for test runs. The package offers the option to match one (for deduplication) or two files. In the post-processing phase, records can also be grouped, and 'possible' and 'excluded' matching candidates can be manually examined and processed.
  - *Method:* It uses a matching method with adjustable weights with specific probabilities (based on Fellegi and Sunter).
- **SSA NAME3** *(Search Software America; www.searchsoftware.com).* With this package, it is possible to match files with people, businesses and other identifiers. It is a commercial package embedded in a system with components to improve and present business information.
  - *Package:* This package also has routines to cleanse and parse data in the pre-processing phase, including standardising the keys and creating subsets (blocking). The routines must, however, be built into existing software. The package offers the option of aggregating a set of records, for example, people or business units, into other units, for example, household and Enterprise. It works both online and in the batch.
  - *Method:* The matching is based on a method with weights.
- **IQ-Matcher** (Intech Solutions; www.intechsolutions.com.au).
  - *Package:* This package offers – just as other commercial packages mainly aimed at the totality of business information – options for cleansing, standardising and matching data. It supports deduplication, and can process large files and multiple files.
  - *Method:* The matching method is a method with weights (probabilities).
- **AutoMatch** (part of Integrity; www.vality.com).
  - *Package:* There are various options to check and standardise data (pre-processing phase). Blocking is possible. There are different options to compare strings.
  - *Method:* Matching based on weights (probabilities; Fellegi and Sunter).
- **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.

[23] In their own words: 'Gartner, Inc. is the world's leading information technology research and advisory company.' See: http://www.gartner.com/technology/home.jsp

Finally, it is also possible to develop your own individual *customised* packages. However, this requires also maintenance. Such pacakages are often specifically geared towards a certain approach and method. The flexibility of such packages tend to be limited.

Because statistical or synthetic matching falls outside the scope of this paper, we did not examine software that supports this method.

8.2 IT considerations

Adding file matching into statistical or other processes requires setting up an information architecture that fits in with the existing architecture. In the case of Statistics Netherlands, aspects to be considered are: saving the results in the DSC so that others can use the interim and final results, and utilising standard software instead of its own customised software.

Even though software is required for matching data, this does not discharge users from the obligation of having a good understanding of the procedure and knowing what they are doing. Matching can be a complex process. However, a situation must be prevented in which the matching process is a 'black box' for the user. This risk is more present when using commercial packages, because the seller is not always open about the actual source code.

It is important to ask the right questions when selecting matching software. Appendix C contains a series of considerations for this purpose.

9. Special subjects

In this chapter, we examine several special subjects that play a role in the use of matching in practical situations, or those that can be resolved using the matching methods described in this document.

9.1 Large files

This creates problems when determining the matching candidates, because there is a very large number of record pairs that must be examined. If we have two files, say A and B, then the number of records that will have to be compared is $|A||B|$, that is, the number of records in A multiplied by the number of records in B. If A and B each have a number of records on the order of 10^5, then the search space will be on the order of 10^{10} (= ten billion) pairs of records. An obvious method to deal with this situation is to drastically reduce the search space (the number of records to be inspected). One way to do this is to partition the matching files using, for example, a matching (or other) variable, the so-called *blocking variable*. Techniques from data mining are of interest for this purpose. However, this involves a relatively recent expansion of matching theory, the value of which must be demonstrated by further research.

9.2 Determining matching parameters

When using matching based on secondary keys, several problems arise that have already been discussed, but we want to emphasise them again here. When working with matching weights, it is also important to find the right cut-off values. Otherwise, all the records from one matching file can potentially be matched with all the records from the other file. It is important to find a suitable way to indicate when the matches are too weak for further serious investigation. The other problem concerns determining a single (multivariate or other) metric if a matching key consists of several secondary keys, each with its own metric. The problem here involves using weights to combine the metrics-per-secondary key variable into a single metric. Part of this problem is addressed in Section 7.3.1.1.

9.2.1 Cut-off values for matching weights

Chapters 6 and 7 include a discussion of cut-off values that can be used when determining candidate matches if matching weights are used. Because it can be difficult to establish a good cut-off value without further information, it must first be assumed that some experimentation must take place with several test matches to arrive at a suitable cut-off value. If the cut-off value is known, then you can include the pairs of records in the MC graph for which the weights, for example, are above this cut-off value. (In this case, do not include the weaker potential matches.) If you choose a cut-off value that is too large and want to experiment with a smaller cut-off value, then you will have to recalculate the MC graph. This is not convenient. In that case, it is better to calculate the MC graph for the smallest cut-off value that you want to examine. For larger cut-off values, you can make use of the calculated MC graph with this (smallest) cut-off value.

9.2.2 Weights in metrics for composite matching keys

Chapter 7 sets out how you can derive a multivariate metric by taking the weighted sum of the metrics per variable. For all values $a_i > 0$ for $i = 1,...,n$, you will formally obtain a metric in this way. The question that may arise is how you properly select the a_is for a specific case. We cannot answer this question just like that; more research is needed. In practice, this also means that you will have to experiment with the data; empirically you should determine which combination produces good results. You must also pay attention to the quality of the scores per variable. The quality is not necessarily the same for the variables in a single file, and in particular for the secondary keys.

9.3 Matching related units

Up to now we have assumed so far that the aim of matching is to match information in different files pertaining to the same units. In practice, this situation can be more complicated than this, however. To illustrate: composite units, such as businesses, may transform over time: they may split or merge with other similar units and form new businesses. It is also possible that the information from two matching files relates to two different reference times that are far enough apart for mutations in the units to become apparent. These mutations are a result of the dynamics that are present in many[24] populations. If a unit in one matching file is still present as such, but the split-off parts (or some of them) are present in the other file, then it is not useful to look for the same units: they are simply not there. To the extent that reference times are farther apart, this effect becomes stronger. Furthermore, other effects resulting from the dynamics of the target population play a more prominent role: new units enter ('birth'), or exit ('death'). These effects also play a role in populations consisting of 'atomic' units, in other words, non-composite or simple units, such as people. If matching is carried out using primary keys, this dynamics does not produce matching problems, provided that it is known how the composite units have evolved from one another. But if the matching is based on secondary keys, a complication may arise. One must then be prepared that the same units are not necessarily paired with one another, but related units may also sometimes have to be matched, namely units that have emerged from other units. Matching is possible if there is a matching table, where the relationships between the between the units in both files and over time are recorded, possibly with the events that have led to the mutations.

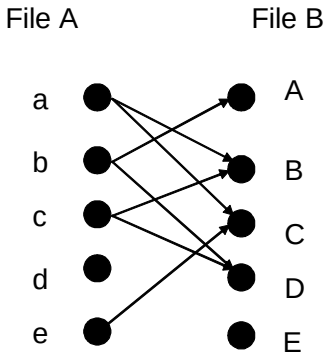
A bipartite graph is not suitable to illustrate a matching situation as referred to in this section. Instead, a bipartite directed graph, or a bipartite digraph for short, will be used. An example of this can be found in Figure 9.1.

---

[24] This dynamics occur in all populations discussed in this document. This concerns populations of 'living' objects, which change continuously. Static populations do exist, such as the 'population' of works by a composer from the baroque period. However, even in this situation, there is still a chance that new works by this composer will be discovered, or that a work which was originally attributed to this composer will later be revealed to have been composed by someone else. But this concerns our knowledge of populations and not the populations themselves. This knowledge may also be dynamic. In practice we are always dealing both with a population and with our knowledge of this population. In temporal databases both aspects are taken into account. In the present paper, however, we will not dwell on this subject.

MC digraphs can be used as matching digraphs, but only in situations where there is asymmetry, for example, if the files to be matched relate to two different points in time or periods. The direction of the arrows can indicate the development, so is in the direction of the 'time flow'. Instead of a relationship that can be described as 'is the same unit as', as is standard in matching, we can also describe a relationship as 'has arisen from the unit'. The same as for the edges in MC graphs, the arrows in MC digraphs may be accompanied by matching weights.

Figure 9.1 Example of two files with different reference times and the relationship between them



Interpretation: a splits into B and C. Or: unit a in file A may have evolved into unit B or unit C in file B. Please note: arrows always point from points in file A to points in file B.

9.4 Dealing with 'remainders'

In practice, it can be a problem to simply leave out records that did not match, because this can make all sorts of figures inconsistent. In such cases, you can decide to match all or part of the non-matched records with one other. This is possible even if there is a high risk that the matches obtained in this way do not describe the same units. You would then find yourself in a situation comparable to statistical matching, in the sense that it is very likely that the units matched will be different, but they will have certain properties in common. Another option is to work with matching weights, and to establish lower cut-offs such that more matchable records remain. It is possible that this will produce an entirely different solution than that obtained previously. If that is not desired, we must try to match the remaining records separately from the matched records. This could mean that, in the process, considerable compromises must be made.

9.5 Matching personal details

Here we discuss an example of a matching situation and how this should be dealt with. Suppose we have a register R with information about persons, with a primary key in addition to secondary keys.

We also have two files A and B with personal details. A and B do not have a primary key, but secondary key variables, such as name, address, age, etc. We assume:

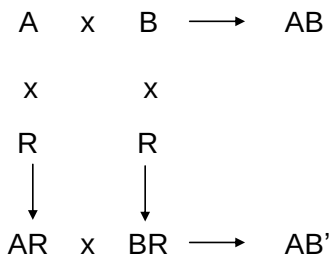
- A and R can be matched based on common secondary key variables
- B and R can be matched based on common secondary key variables
- A and B can be matched based on common secondary key variables

The common variables do not have to be the same in the three situations. To match A with B, different options are possible.

1. A is matched directly with B, making use of the common secondary key variables in the two files. This produces file AB. In notation: $A \times B \rightarrow AB$
2. A is matched with R, and B is matched with R, each based on common variables. The matching files AR and BR are then matched based on the primary key from R. This last match is therefore a join. This produces matching file AB'. In notation: $A \times R \rightarrow AR$, $B \times R \rightarrow BR$ and $AR \times BR \rightarrow AB'$.

Figure 9.2 depicts these matches in diagram form. For the sake of the summary, the details about the separate matches have been left out, in particular with regard to the matching variables used.

Figure 9.2 Two options of matching files A and B: directly or indirectly through register R



The indirect matching through files AR and BR is a join. In the other cases secondary keys have to be used.

Without further knowledge, it cannot be established which matching file is better, AB or AB'. However, it is interesting to use both matching methods in an experiment and to compare the resulting files. In general, it is not possible to say which of the two matching methods works best in the experiment (and possibly also in . Which ones are possible at all depends on the available common matching variables.

9.6 Matching business data

9.6.1 Specific aspects

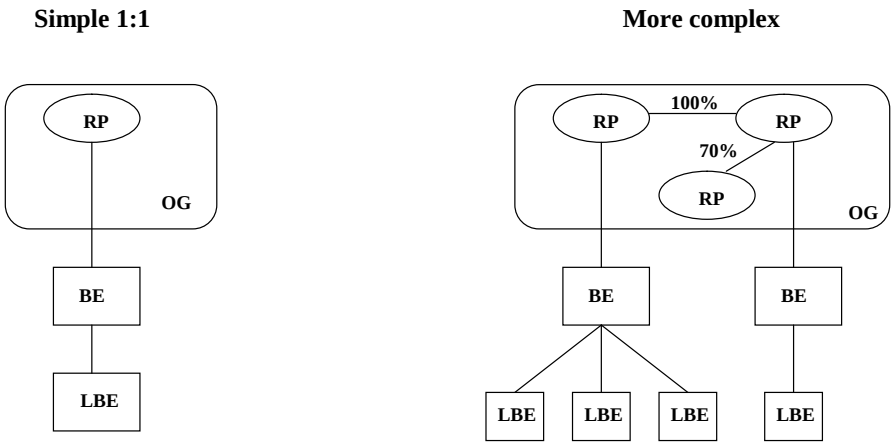
In the matching literature, the background and the examples are usually based on matching personal details. While comparable problems arise when matching business details, there are also clear differences. When matching business details, you must take account of the following and other aspects:

- An enterprise is a construct. As such, it is not always clear what is part of it and what is not. The composition and meaning of, for example, the statistical unit 'Enterprise Group' may be different from the composition and meaning of an Enterprise for the Tax Administration or for the outside world[25]. This could mean that you think you are matching two of the same units, whereas these are actually different constructs.
- At two reference times, the identifying characteristics of an enterprise may be the same in two files, and therefore they can be matched directly. However, between these two reference points, any number of events (for example, a merger or split) could have taken place, as a result of which the data from the same enterprise at these two different reference times can no longer be compared. This means that mutually comparing the variables from the different matched files is no longer as evident.
- If you want to look at the development an enterprise over time by building a time series, the problem can actually be that it still concerns the same enterprise, but that the identifying variables, for example, the name of the legal entity, are not the same at the different reference times.
- The identifying characteristics based on the name of the same enterprise can differ greatly. Examples of this are the trade name, the name of the owner, the name of the legal entity, the name of a part of the enterprise (the production unit), a generally used name (for example, in advertising) and the name of the accountant. The same is true for the address. This could be the postal address or the visitor's address, the address of the head office or the address of a branch, or the address of the accountant.
- The data as collected by businesses is strongly related, much more than for persons. This can mean that calculating, for example, productivity, from two matched files, one containing the numer of persons employed and the other the turnover, can be risky. You must be very certain that it concerns the same construction of the enterprise and that the values used were measured at virtually the same reference time.
- A single business may conduct multiple economic activities (see the NACE). When matching files, this can lead to problems, because it is not always equally clear which activities are included in the different matched files, and which are not. An example is the way businesses position their private pension funds, sometimes in and sometimes outside the business. Another related problem often arises in functional statistics, where you want to relate data from different files to that one functional activity. This is usually not possible because, for example, the turnover, costs, etc., are stated for the entire business and are not broken down to activities within the business..

[25] For example, more or fewer control relationships may be included.

- Business Units can be combined into a larger unit such as the Enterprise Group. Data from Business Units cannot always simply be added up to and compared with the data from the Enterprise Group, For example, the Enterprise Group may involve a consolidation, meaning that the mutual deliveries and flows between the smaller Business Units are not included in the totals of the Enterprise Group.. So, the sum of the Business Units does not always have to be the same as the consolidated total.
- The administrative processing of events and mutations in businesses often progresses much slower than mutations in personal details, as a result of which files can still be 'polluted' with old data. A difference can also arise between the point in time when the event occurred and the point in time when it was recorded in the administrative system and processed in files.
-

Figure 9.3 Different statistical units for business statistics



RP = Legal Entity
OG = Enterprise Group
BE = Business Unit
LBE = Branch (Local Business Unit)

9.6.2 The main business unit for statistics

When matching business data, there are several related constructions or units that can represent the business in one way or another. This concerns – mainly in terms of statistics – the following units: (1) The Legal Entity (known as a "CBS person" at Statistics Netherlands; this can be both a legal entity and a natural person), an Enterprise Group , a Business Unit and a Local Unit or Branch. Roughly speaking, the Enterprise Group is the central unit. An Enterprise Group is composed of one or more Legal Entities. If there is more than one Legal Entity, then there must be control relationships between these Legal Entities. This does not have to be 100% control. These control relationships are based on information from the Chamber of Commerce and the Tax

Administration. The Enterprise Group is the financial actor in the economic process. The Legal Entities and therefore the Enterprise Group can represent one or more production units, called Business Units . This concerns the productive actor in the economic process. At Statistics Netherlands, most business statistics are based on the Businees Unit. Business Unitss can have multiple local units or branches at different locations.

9.6.3 Developments

Among researchers, there is an increasing need to follow enterprises over time (economic demography). To this end, time series must be set up in which, for each event, it is clear which units in a new situation have arisen from units from the old situation, or which have possibly been newly created or ended. This concerns not only 1:n or n:1 relationships, but also n:m relationships. This makes the situation complex. In addition, it is important to record both the event and the time when the event occurred. The complexity increases if the number of variables to which this series relates is increased over time. Consequently, both ‘horizontal’ and ‘vertical’ matching must take place.

Another development is that there is an increasing need to match personal and business details. For example, to look at what type of people work where. This type of matches requires that there is a primary key to be matched in the file with the ‘largest unit’, in this case the business, and a foreign key to be matched in the file with the smallest unit, in this case the employee.

10. References

De Jong, W.A.M. (1991), *Technieken voor het koppelen van bestanden*. Statistical studies, M41, SDU/publishers/ Statistics Netherlands publications, The Hague.

D'Orazio, M., di Zio, M. and Scannu, M. (2006), *Statistical matching*. Wiley, New York.

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

Gartner (2007), *Magic quadrant for data quality tools 2007*. Gartner RAS, research note, June 2007.

Gill, L. (2001), *Methods for Automatic Record Matching and Linking and their use in National Statistics*. National Statistics Methodological series no. 25, Oxford University.

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

ISAD (2008a), *State of the art on statistical methodologies for the integration of surveys and administrative data*. ESSnet Statistical Methodology project on the integration of survey and administrative data, a CENEX project.

ISAD (2008b), *Recommendations on the use of methodologies for the integration of survey and administrative data*. ESSnet Statistical Methodology project on the integration of survey and administrative data, a CENEX project.

Lenz, Rainer (2003), *A graph theoretical approach to record linkage*. Paper for the joint ECE/Eurostat worksession on statistical confidentiality 17-19 April 2003.

Mardia, K., Kent, J. and Bibby, J. (1982), *Multivariate analysis*. Academic Press.

Nemhauser, G.L and Wolsey, L.A. (1988), *Integer and combinatorial optimization*. Wiley Interscience.

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

Newcombe, H.B., Kennedy, J.M., Axford, S.J. and James, A.P. (1959), Automatic linkage of vital records. *Science* 130, 954-959.

Papadimitriou, C.H. and Steiglitz, K. (1998), *Combinatorial optimization*. Dover.

Sankoff, D. and Kruskal, J.B. (eds.) (1983), *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley.

Statistics New Zealand (2006), *Data integration manual*. Statistics New Zealand, Wellington.

Van de Laar, R. (2008), *Conceptuele typering van processtappen naar businessfunctie*. Internal report, Statistics Netherlands, Voorburg.

Wikipedia, article about the EM algorithm, http://en.wikipedia.org/wiki/EM_algorithm.

Wikipedia, article about Record linkage, http://en.wikipedia.org/wiki/Record linkage.

Willenborg, L. and De Waal, T. (2000), *Elements of statistical disclosure control*. Lecture notes in statistics, Vol. 155, Springer.

Winkler, W.E. (1985), *Exact matching lists of businesses: blocking, subfield identification and information theory*. Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 438-443. Published in an extended version in Alvey W., Kills B. (eds.) Record Linkage Techniques - 1985, Proceedings of the Workshop on Exact Matching Methods}, pp. 227-241.

Winkler, W.E. (2006a) *Overview of Record Linkage and Current Research Directions*. U.S. Bureau of the Census, Statistical Research Division Report Series, n.2006/2.

Appendix A. The Fellegi-Sunter model

The Fellegi-Sunter model (1969) is based on a decision theoretical approach, which formalises a procedure from Newcombe (see Newcombe et al., 1959). Because the Fellegi-Sunter method has had and continues to have a significant influence on the world of matching, we will describe it here briefly, using De Jong (1991).

We assume two sets with units A and B, both associated with a population P. A typical element of A is a, and for B, this is b. We assume that there are elements that occur in both populations, so $A \cap B \neq \phi$. We denote the records corresponding with the units in A and B using $\alpha(a)$ for $a \in A$ and using $\beta(b)$ for $b \in B$. We denote the files associated with A and B as $\alpha(A)$, $\beta(B)$. It is important to distinguish between the sets of units (A and B) and their representations in the form of files ($\alpha(A), \beta(B)$). A distinction must also be made between the representations α and β, because they can differ from one another. The representations include errors in how they are stated, processing errors, etc.; in short, all non-sampling errors.

It is possible, for example, that the first name of the same person in one file is represented as 'Hugo' (given name) and the other as 'Huug' (nickname). This is an example of a unit a where $\alpha(a) \neq \beta(a)$. Even so, it is possible that there are two different units a, b (therefore $a \neq b$) where $\alpha(a) = \beta(b)$.

We now consider two important subsets of the set of record pairs from $\alpha(A)$ and $\beta(B)$: $\alpha(A) \times \beta(B) = \{(\alpha(a), \beta(b) \mid a \in A, b \in B\}$:

1. The *matched set*: $M = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), a = b\}$.
2. The *unmatched set*: $U = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), b \in \beta(B), a \neq b\}$.

These sets M and U are not known in practice.

The comparison vector $\gamma = (\gamma_1, ..., \gamma_K)$ associated with the records in the two files is defined as follows:

$$\gamma(\alpha(a), \beta(b)) = (\gamma_1(\alpha(a), \beta(b)), \gamma_2(\alpha(a), \beta(b)), ..., \gamma_K(\alpha(a), \beta(b))),$$

where each γ_i ($i = 1, ..., K$) symbolises a specific comparison. For example, γ_1 may be an indicator that records a correspondence in the gender of two people (same gender yes/no). γ_2 could be an indicator of two surnames that are/are not the same. And so forth. Then let $\Gamma = \{0,1\} \times ... \times \{0,1\}$ (K times) be the set of possible realisations of γ. Based on γ, a record pair must be classified as belonging to M or U. Two records are matched if the record pair is classified as belonging to M. The set of record pairs $\alpha(A) \times \beta(B)$ therefore divides into two sets

3. The *matched set* (set of links):
   $L = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), \alpha(a)$ and $\beta(b)$ are matched$\}$.
4. The set of *non-matched records* (non-links):
   $N = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), \alpha(a)$ and $\beta(b)$ are not matched$\}$

In the matching, efforts are made to make L as similar as possible to M (or, equivalently, N to U). This boils down to making an effort to avoid mismatches and missed matches. The following is true:

$$\{\text{missed matches}\} = \{(\alpha(a), \beta(b)) \mid (\alpha(a), \beta(b)) \in M \setminus L\}.$$

$$\{\text{mismatches}\} = \{(\alpha(a), \beta(b)) \mid (\alpha(a), \beta(b)) \in L \setminus M\}.$$

Fellegi and Sunter (1969) use as matching criterion the ratio

$$R(\gamma) = m(\gamma)/u(\gamma),$$

where

$$m(\gamma) = P[\gamma(\alpha(a), \beta(b)) = \gamma \mid (\alpha(a), \beta(b)) \in M],$$

the fraction of record pairs is $\alpha(A) \times \beta(B)$ in M with score vector γ, and

$$u(\gamma) = P[\gamma(\alpha(a), \beta(b)) = \gamma \mid (\alpha(a), \beta(b)) \in U],$$

the fraction of record pairs $\alpha(A) \times \beta(B)$ in U with score vector γ. A practical problem is that $m(\gamma)$ and $u(\gamma)$ are not known. However, in the first instance, they can be approximated.

Ideally, we would want to be able to establish one parameter c such that

If $R(\gamma) \geq c$, then $\alpha(a)$ and $\beta(b)$ are matched,
If $R(\gamma) < c$, then $\alpha(a)$ and $\beta(b)$ are not matched.

In practice, with one parameter, it is often not possible to separate M and U. For this reason, it is better to work with two cut-off values $c \leq d$ such that

If $R(\gamma) > d$, then $\alpha(a)$ and $\beta(b)$ are matched,
If $R(\gamma) < c$, then $\alpha(a)$ and $\beta(b)$ are not matched,
If $c < R(\gamma) < d$, then $\alpha(a)$ and $\beta(b)$ are considered as provisional matches.

A special computer programme is needed to eliminate record pairs using a γ, for which $R(\gamma) < c$ and to find the record pairs for which $c < R(\gamma) < d$, which will then have to be inspected by a matching specialist to determine which are matches and which are not. This inspection is very time consuming and expensive, and we want to keep it to a minimum. This is in fact the goal of the Fellegi-Sunter method.

We will end our discussion here of the method developed by Fellegi and Sunter. The basic idea of their approach should now be clear. It should also be clear that a variety of estimates must still be made to actually use this method in practice. A problem is that the sets M and U are not known, therefore neither is $R(\gamma)$ for a given γ. For this kind of issues, we refer interested readers to the original article by Fellegi and Sunter. See also De Jong (1991) or Herzog et al. (2007, Chapter 9).

Appendix B. More about metrics

Here we want to continue the discussion that stopped at the end of Section 7.3.1.1. We will present an elaboration here that is important in practice, but rather specialised and technical.

To continue with our discussion in Section 7.3.1.1: we examine the problem of text, in this case, names, which can be spelled incorrectly because, for example, they are written based purely on the pronunciation. In that case it is impossible (in Dutch) to know whether someone's last name is 'Jansen', 'Janssen', 'Janszen', etc.[26] We would like to associate such names with one another by matching the names based on a phonologically inspired code.

In this case, we have a set N (of names with many spelling variations) and a set P of codes based on the pronunciation (such as Soundex)[27] and a function $f : N \to P$ that we can assume to be surjective.[28] This f introduces an equivalence relationship $\approx$ on N, where $n_1 \approx n_2$ for $n_1, n_2 \in N$ if $f(n_1) = f(n_2)$. Using this f, we can introduce a metric d_L^* on P, derived from d_L, and the metric on N, as follows: for $a, b \in P$, $f^{-1}(a) \subseteq N$ and $f^{-1}(b) \subseteq N$ are the non-empty (f is surjective!) complete inverse images of a and b, where $f^{-1}(a) \cap f^{-1}(b) = \phi$ if $a \neq b$. Then

$$d_L^*(a,b) = d_{Haus}(f^{-1}(a), f^{-1}(b)) = \max\{\sup_{x \in f^{-1}(a)} \inf_{y \in f^{-1}(b)} d(x,y), \sup_{y \in f^{-1}(b)} \inf_{x \in f^{-1}(a)} d(x,y)\},$$

where the last equality represents the so-called Hausdorff distance d_{Haus} based on d. Here 'sup' and 'inf' represent 'supremum' and 'infimum' respectively, which we could suitably replace in our context by 'maximum' and 'minimum' (because, in practice, we are dealing with finite sets of words, names, etc.). We therefore find that:

$$d_L^*(a,b) = \max\{\max_{x \in f^{-1}(a)} \min_{y \in f^{-1}(b)} d(x,y), \max_{y \in f^{-1}(b)} \min_{x \in f^{-1}(a)} d(x,y)\}$$

This is a metric on 2^N, the collection of subsets of N, if at least d is limited, which means that $d(x,y) \leq M$ for all $x, y \in N$ for some $M > 0$[29]. This metric is defined as follows for sets $A, B \subseteq N$:

$$d_{Haus}(A,B) = \max\{\sup_{x \in A} \inf_{y \in B} d(x,y), \sup_{y \in B} \inf_{x \in A} d(x,y)\}.$$

---

[26] Verification by the person whose name it is could ensure this problem is avoided. However, we assume that this is not possible here. For proper names, verification is self-evident. However, for a text that describes, for example, someone's profession, an interviewer is probably not as inclined to display his/her own spelling ability (or lack thereof).

[27] Based on set of rules, a code is derived from a given string. The document on coding in the Methods Series provides the rules for the Dutch version of Soundex.

[28] If not, then we limit the range (codomain) of f to $f(P) \subseteq P$. $f : P \to f(P)$ is surjective.

[29] Otherwise, it is not a metric but an 'extended metric', which can also take the value of ∞, i.e. infinity.

Ideally, the spelling variations of a word should end up in the same equivalence class, and then have the distance 0 between the two. For this, one could make use, for example, of the Soundex algorithm (for Dutch).

In Herzog et al. (2007, Chapter 13), 'comparator metrics' are introduced which are intended for strings with typographical errors, such as transpositions (switching two subsequent characters, such as 'carpentre' instead of 'carpenter'). One is from Jaro and the other from Winkler. Without further motivation, we present them below. For more information, see Winkler et al. (2007). The Jaro similarity measure is as follows for two strings s and t:

$$d_J(s,t) = w_s \frac{c}{l_s} + w_t \frac{c}{l_t} + w_{transpos} \frac{c-\tau}{c},$$

where:

- w_s is the weight associated with the s, w_t is the weight associated with the t, $w_{transpos}$ is the weight associated with the transposition, $w_s, w_t, w_{transpos} > 0$ and $w_s + w_t + w_{transpos} = 1.$
- $c > 0$ is the number of characters that s and t have in common, for which the distance between the common characters is less than the length of the shortest string (i.e. $\min\{l_s, l_t\}$).
- l_s is the length of s, l_t is the length of t.
- τ is the number of transposed characters.

If $c = 0$ then $d_{Jaro}(s,t) = 0.$

The Winkler similarity measure is defined as follows:

$$d_W(s,t) = d_J(s,t) + 0.1 \bullet i \bullet (1 - d_J(s,t)),$$

where $i = \min\{j, 4\}$, and j is the number of characters that the strings s and t have in common at the start.

Another way of comparing strings is to divide them into trigrams and to count how many trigrams match with each other, and to what extent. Here we use the order of the trigrams in the string. We also use this order for the match. In the match, we count the number of characters that correspond and which are in the same position, and the number of characters that are the same, regardless of their position in the trigram.

Appendix C. Considerations when selecting matching software[30]

General:

1. Is the seller a reliable and solid company? Can it provide technical support? What is the seller's vision in the longer term?
2. How well is the system documented? Can users teach themselves to use the system, or do they need support (a little or a lot)? Is training provided?
3. Is there a user group?
4. How quickly can an average matching project implement the system?
5. Can the package and the matching process be adapted (easily) to the specific wishes of the user?

Data and system management:

1. On what platform can the software be used? And what are the hardware requirements?
2. How well does the package fit in with the actual software and databases that are used?
3. What data format and storage does the package allow? Does the format fit in with the existing information architecture?
4. Is it a complete system ('out of the box' matching), or a set of components around which a system must still be built? How complete is this set of components?
5. Is it a single-user or multi-user package?
6. Can the system work interactively or only in the batch?
7. What is the maximum size of the files (number of records) that the package can process, including the resulting file with matching candidates?
8. How does the package process the records (for example, temporary file, sorted or based on pointers)?
9. Does the package support functions to examine and manipulate the data? Does it do so during the interim steps or only at the end of the process?
10. What does the interface of the package look like?
11. How well does the software perform, especially with regard to large files and more advanced methods?

Costs:

1. What are the purchase and maintenance costs (for example, for connections with other software or a database and upgrades) of the package?
2. How much does a licence or site licence cost?
3. How much does it cost to train people to use the package?

Pre-processing phase:

1. What options are present for the pre-processing phase? Are there options to check, edit and standardise variables (or groups of variables or related variables)? To what extent can users themselves define actions on variables and records? Can these actions also be performed on groups of records (under conditions to be defined by the user)?
2. Is it possible, for example, to 'take apart' postal codes?

[30] See C. Day, in Record Linkage techniques, chapter 13, A checklist for evaluating record linkage software

3. Do the operations result in a new file (the old values are no longer available) or are extra variables added?
4. Are there possibilities for deduplication?
5. Can you use blocking variables? Is it possible, per run, to define more than one blocking variable (and different comparison methods)?
6. Can the user define subsets of the file based on which matching must be performed? Can records or groups of records be set aside (temporarily)?
7. Does the package support the selection of smaller 'test files' to perform a test run?
8. Are there options to define 'commentary fields' (before and after)?

Matching methods:

1. What matching methods are supported?
2. What type of matches does it allow, such as one file with itself (deduplication), two files, more than two files, including matching with reference files?
3. Is it generic software, or does it only focus on specific uses (for example, matching based only on names or only for personal details)?
4. Can matching take placed based on people and businesses, or other identifiers?
5. Is the matching process a 'black box' or can the user guide it using parameters? If yes, is it easy to make a file with parameters?
6. How many variables may be included in the matching key? Can the system also deal with foreign key matches?
7. Is the package suitable for matches based on related units ('has arisen from' and 'originates from')?
8. Can the user define the matching variables and the desired comparisons?
9. What comparisons of the keys and the variables of keys are possible? For example: character-by-character, Soundex (phonetic), string comparisons, metrics, comparisons of postal codes (with the numeric and alphanumeric part, for Dutch postal codes), date and time comparisons, comparisons based on conditions to be determined by the user?
10. Is it possible to use metrics, probabilities, cut-off values (does the system help by making suggestions to establish the cut-off value, or does it provide information for this)?
11. How does the package deal with missing elements in the key variables (such as a weight 0), even if a method with weights is used?
12. Can users, for example, specify critical variables, for which they have established that records only match if these variables (of the key) correspond? In the comparison, is it possible to differentiate per variable?
13. Can weights be calculated per variable and for the total, and which methods are available for this?
14. When matching, is it also possible to take account of the dependencies between variables?

Post-processing phase:

1. Does the package support options to make estimations of Type I and Type II errors (quality of the matches)?
2. What options are present to generate summaries, such as to perform an evaluation (with, for example, matched and non-matched records, doubtful cases, calculated weights, etc.)? Is it possible to adapt the format and the content of the summaries? Are there summaries in graphic form?

3. Are there options to generate statistics to evaluate the process?
4. Is it easy to compare the results of several different runs?
5. In what formats can the results of the match be recorded and stored?

Version history

Version	Date	Description	Authors	Reviewers
Dutch version: Koppelen				
1.0	02-03-2010	First Dutch version	Leon Willenborg Nico Heerschap	Fred Gast Rob van de Laar Sander Scholtus
English version: Matching				
1.0E	14-02- 2012	First English version	Leon Willenborg Nico Heerschap	