Koppelen

10

Leon Willenborg en Nico Heerschap

Statistische Methoden (10013)

Centraal Bureau voor de Statistiek

Den Haag/Heerlen, 2010

Verklaring van tekens

.	=	gegevens ontbreken
*	=	voorlopig cijfer
**	=	nader voorlopig cijfer
x	=	geheim
–	=	nihil
–	=	(indien voorkomend tussen twee getallen) tot en met
0 (0,0)	=	het getal is kleiner dan de helft van de gekozen eenheid
niets (blank)	=	een cijfer kan op logische gronden niet voorkomen
2008–2009	=	2008 tot en met 2009
2008/2009	=	het gemiddelde over de jaren 2008 tot en met 2009
2008/'09	=	oogstjaar, boekjaar, schooljaar enz., beginnend in 2008 en eindigend in 2009
2006/'07–2008/'09	=	oogstjaar, boekjaar enz., 2006/'07 tot en met 2008/'09

In geval van afronding kan het voorkomen dat het weergegeven totaal niet overeenstemt met de som van de getallen.

Colofon

Uitgever
Centraal Bureau voor de Statistiek
Henri Faasdreef 312
2492 JP Den Haag

Prepress
Centraal Bureau voor de Statistiek - Grafimedia

Omslag
TelDesign, Rotterdam

Inlichtingen
Tel. (088) 570 70 70
Fax (070) 337 59 94
Via contactformulier: www.cbs.nl/infoservice

Bestellingen
E-mail: verkoop@cbs.nl
Fax (045) 570 62 68

Internet
www.cbs.nl

ISSN: 1876-0333

6016510013 X-37

Inhoudsopgave

1. Inleiding op het thema

1.1 Introductie en achtergrond

De toenemende vraag naar tijdige, gedetailleerde en kwalitatief goede statistieken enerzijds, en de plicht om zoveel mogelijk gebruik te maken van bestaande administraties anderzijds, dwingt om te kijken naar alternatieve wegen om statistieken te produceren, bijvoorbeeld door informatie uit verschillende bestanden onderling te koppelen. Zo zijn administraties niet ingericht om statistieken te maken. Om toch de gewenste statistieken te kunnen produceren is het nodig om administraties en eigen onderzoek aan elkaar te koppelen tot beter bruikbare datasets. Daarbij moet ook worden gedacht aan longitudinale reeksen. Aan de outputkant bestaat meer behoefte om verschijnselen in hun onderlinge verband te presenteren en niet alleen als losse statistieken. Dat maakt het mogelijk om over bredere thema's te publiceren en nieuwe output te ontwikkelen. Voorbeelden hiervan zijn: de thema's over veroudering en globalisering. Hiermee kan beter aan de actuele behoeften van gebruikers van statistieken worden voldaan.

Het koppelen van gegevens draagt onder meer bij aan:

- het sneller kunnen publiceren van (nieuwe) output;
- een betere kwaliteit van de data, door bijvoorbeeld onderlinge confrontatie;
- het verminderen van de enquêtedruk en dus minder kosten bij de respondenten en berichtgevers;
- het verminderen van de kosten van het CBS omdat geen eigen onderzoek meer hoeft te worden uitgevoerd.

Het koppelen van gegevens ondersteunt daarmee de belangrijkste doelen van het CBS, zoals nieuwe output, minder enquêtedruk, beter gebruik van administratieve bronnen en lagere kosten. In welke mate het (meer) koppelen van gegevens bijdraagt aan meer efficiëntie, in de zin van minder vte's, is moeilijk te bepalen. Enerzijds zal het leiden tot besparingen als het bijvoorbeeld gaat om het verder beperken van eigen onderzoek. Anderzijds vraagt het koppelen van bestanden en het analyseren van de resultaten ook om extra capaciteit. Duidelijk is wel dat het vraagt om andere competenties. Ook het opbouwen van de kennis van de te koppelen bestanden is van belang en vergt vaak veel inspanning en capaciteit.

Dit themarapport beschrijft de *methodologie van het koppelen*. Dit betreft vooral de problematiek van het bij elkaar brengen van informatie van records afkomstig uit twee of meer bestanden die betrekking hebben op dezelfde eenheden, waargenomen op vrijwel hetzelfde tijdstip. Dit kan een betrekkelijk eenvoudige opgave zijn, namelijk als er een gemeenschappelijke en eenduidige koppelsleutel is voor de eenheden in beide bestanden, waarbij de scores van de koppelsleutel bovendien betrouwbaar zijn. Bij personen kan men denken aan een koppelsleutel als het Burgerservicenummer (*BSN*). Het kan echter ook veel lastiger zijn, bijvoorbeeld als zo'n eenduidige koppelsleutel niet bestaat, maar wel een aantal (secundaire) koppelvariabelen zoals naam, adres, geboortedatum en leeftijd (op een bepaalde dag) die gemeenschappelijk voorkomen, maar die niet altijd even betrouwbare scores hoeven te hebben. Of het geval dat niet precies dezelfde variabelen in de koppelsleutel voorkomen, maar soortgelijke, bijvoorbeeld met een iets ander domein. Nog lastiger is het als ook de eenheden in beide bestanden niet hetzelfde zijn, bijvoorbeeld als gevolg van de dynamiek in populaties. Naast geboorte en sterfte van eenheden,

kunnen eenheden verouderen of over gaan in andere eenheden. Voorbeelden daarvan zijn fusies of splitsingen van bedrijven.

1.2 Plaats in het statistiekproces

Koppelen van gegevens beperkt zich niet tot één specifieke plek in het statistische proces. In feite kan op elke plek in het statistische proces wel sprake zijn van het koppelen van gegevens. Aan de inputkant begint het al bij de opbouw van het statistisch kader. Veelal is een combinatie van bronnen nodig om zo'n kader of ruggengraat samen te stellen. Dat geldt bijvoorbeeld voor het Algemeen Bedrijven Register (of de Eenhedenbase) bij de economische statistieken. Daarbij wordt onder meer gebruik gemaakt van gekoppelde gegevens van de Kamer van Koophandel en de Belastingdienst. Bij het verwerkingsproces kan het koppelen van bestanden op verschillende manieren worden ingezet. Bijvoorbeeld als extra informatie bij het controleren van de kwaliteit van de data of bij het afleiden van data, bijvoorbeeld bij het imputeren. Bij de output gaat het vooral om het verkrijgen van nieuwe informatie door het combineren van gegevens uit verschillende bronnen.

1.3 Afbakening en relatie met andere thema's

In dit themarapport worden in eerste instantie koppelmethoden besproken die tot doel hebben gegevens van dezelfde eenheden, maar weergegeven in verschillende bestanden, met elkaar in verband te brengen.

Koppelen is gerelateerd aan andere onderdelen van de Methodenreeks, zoals:

- ***(micro-)integratie*** van gegevens. Daarbij worden gegevens met elkaar geconfronteerd, waarmee allerlei verschillen manifest worden. Deze verschillen dienen verklaard en vervolgens weggewerkt te worden. Het confronteren van de gegevens is slechts mogelijk nadat de bestanden gekoppeld zijn;
- ***coderen.*** Daarbij worden omschrijvingen, die door respondenten in hun eigen bewoordingen gegeven zijn, gekoppeld aan codes uit een classificatie. Hier speelt onder andere het probleem om woorden te kunnen koppelen, wetende dat er spellingsfouten of grammaticafouten gemaakt kunnen zijn of dat synoniemen, hyponiemen of hyperoniemen gebruikt kunnen zijn.
- ***uitzetten van steekproeven.*** Het doel hierbij is om contactinformatie van steekproefeenheden (personen, bedrijven) te koppelen aan interviewers voor het afnemen van interviews. Bij CAPI-interviews gaat het bijvoorbeeld om de woonadressen van personen die in een steekproef zijn getrokken en die bezocht moeten worden door interviewers voor het afnemen van interviews. Bij het toewijzen van adressen aan interviewers wenst men rekening te houden met de maximale interviewcapaciteit van een interviewer en de reisafstanden van interviewers naar woonadressen van de steekproefpersonen. De interviewcapaciteit per interviewer dient gerespecteerd te worden en de reiskosten geminimaliseerd.
- ***disseminatie van gegevens.*** Koppelen van gegevens is noodzakelijk om statistische gegevens in hun onderlinge samenhang te zien en te presenteren.

Afbakening:
Een methode die op het eerste oog een koppelmethode lijkt te zijn, en die bekend staat onder de naam *statistisch of synthetisch koppelen*, is in werkelijkheid een imputatiemethode. De intentie achter deze methode is namelijk verschillend van die van de hier behandelde koppelmethoden. Het

gaat bij statistisch koppelen om het invullen van ontbrekende waarden in een bestand, waarbij een hulpbestand wordt gebruikt. Daar wordt informatie van *soortgelijke* eenheden uit gehaald om de ontbrekende waarden te kunnen invullen. Het gaat dus om soortgelijke eenheden en niet om dezelfde. Om die reden wordt deze methode hier *niet* behandeld. Voor meer informatie, zie D'Orazio, Di Zio en Scannu (2006) en De Jong (1991).

1.4 Afwijking van de standaardindeling van methoden

De bespreking van de koppelmethoden in dit rapport wijkt enigszins af van de standaard bespreking van het onderwerp in de literatuur. Daar wordt een onderscheid gemaakt tussen deterministisch (of exact[1]) koppelen enerzijds en probabilistisch koppelen anderzijds. Onder deterministisch koppelen worden alle methoden geschaard waarbij een ondubbelzinnig voorschrift bestaat wanneer twee records wel/niet matchen. Bij probabilistisch koppelen wordt gebruikt gemaakt van een kansmodel op basis waarvan een koppeling van twee records al dan niet wordt gemaakt. Meer in het bijzonder wordt voor dat laatste gebruik gemaakt van een model voorgesteld door Fellegi en Sunter (1969). Zie Appendix A voor een korte beschrijving van dit model.

Het probleem met deze indeling is dat deterministisch koppelen ook koppelvarianten omvat die evident bedoeld zijn om fouten in de data 'op te vangen'. Bijvoorbeeld als de koppelmethode is om twee records te koppelen als hun scores overeenkomen op 4 van de 5 koppelvariabelen. In onze aanpak is dit een koppelmethode die gebruik maakt van een metriek, die juist wordt gebruikt om rekening te houden met fouten in de data. Probabilistisch koppelen, echter, is in onze aanpak slechts één van de modellen om gewichten te bepalen voor kandidaat-koppelingen.

1.5 Leeswijzer

In dit rapport wordt in hoofdstuk 2 gestart met een algemene beschrijving van wat koppelen is en wat koppelen in de praktijk zo complex kan maken. Het succes van koppelen is niet alleen afhankelijk van de gekozen methode, maar voor een belangrijk deel ook van de getroffen voorbereidingen en nabewerkingen. Daarom wordt in paragraaf 2.3 wat uitgebreider stilgestaan bij de verschillende stappen in het koppelproces.

Voordat wordt ingegaan op de theorie van het koppelen en de koppelmethoden zelf, wordt in hoofdstuk 3 een korte inleiding gegeven van graphen en metrieken, omdat deze een centrale rol spelen bij het beschrijven van de koppelmethoden in de verdere hoofdstukken.

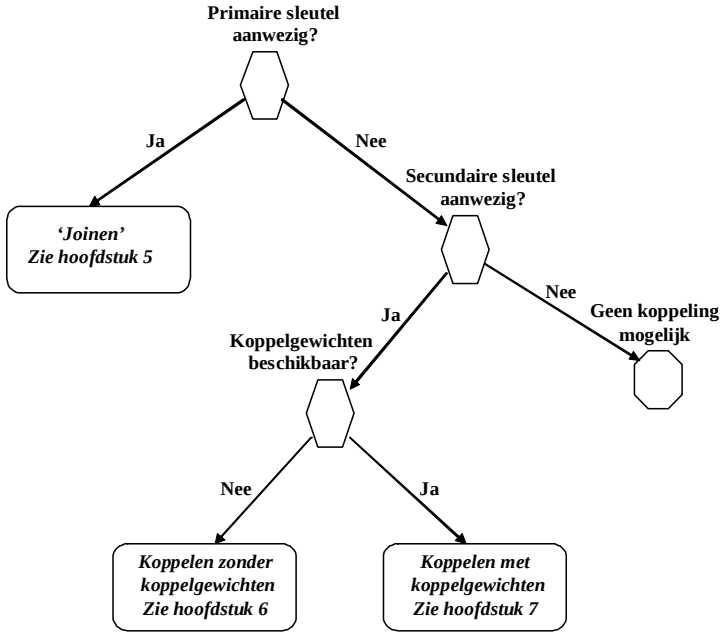
Hoofdstuk 4 gaat in op de theoretische aspecten van het koppelen in meer algemene zin. Via de theorie kan men goed laten zien wat de overeenkomsten en verschillen zijn van de verschillende koppelmethoden. En ook welke verschillende koppelstrategieën er bestaan, en hoe verschillende

[1] Omdat de aanduiding 'exact' zo misleidend is in dit verband willen we hem in dit stuk liever niet gebruiken. Een 'exacte koppeling' suggereert foutloos te zijn. Dat hoeft echter helemaal niet het geval te zijn. Het voorschrift om te koppelen kan wel exact zijn, maar dit voorschrift toepassen op koppelbestanden met fouten in hun koppelvariabelen kan best foutieve koppelingen opleveren. Bovendien werkt probabilistisch koppelen ook met een exact voorschrift. Kortom het begrip 'exact' roept eerder verwarring op dan dat het helderheid schept.

optimaliseringsmodellen kunnen worden geformuleerd. Deze modellen verschillen ten aanzien van de doelen en de randvoorwaarden die men met het koppelen van bestanden kan nastreven.

Daarna passeren in de hoofdstukken 5 tot en met 7 drie methoden van koppelen. Eerst wordt een wijze van koppelen beschreven die typisch is voor databases, maar niet per se daartoe beperkt. Dit gebeurt met behulp van primaire sleutels en wordt ook wel joinen genoemd. Zij is belangrijk omdat zij in de praktijk vaak wordt gebruikt. Het is de eenvoudigste van de koppelmethoden die we in dit rapport behandelen. Daarna worden twee koppelmethoden behandeld die onder minder goede omstandigheden worden ingezet dan bij het joinen. Het gaat daarbij om koppelmethoden die op secundaire sleutels zijn gebaseerd. We maken bij deze methoden het onderscheid tussen koppelen zonder koppelgewichten (hoofdstuk 6) en koppelen met koppelgewichten (hoofdstuk 7). We kunnen koppelen zonder koppelgewichten formeel opvatten als een speciaal geval van koppelen met gewichten (bijvoorbeeld alle gewichten gelijk aan 1). Door gebruik te maken van koppelgewichten wordt de mogelijkheid geschapen potentiële koppelingen onderling in sterkte te scheiden. In figuur 1.1 wordt schematisch weergegeven welke koppelmethoden worden onderscheiden.

Figuur 1.1: Overzicht van de verschillende wijzen van koppelen



Koppelen in de praktijk wordt in de regel softwarematig uitgevoerd. Denk bijvoorbeeld alleen maar aan de grootte van de te koppelen bestanden en de omvang van het aantal koppelkandidaten, die daaruit voortvloeit. In hoofdstuk 8 wordt kort ingegaan op een aantal software pakketten, waarmee kan worden gekoppeld.

Hoofdstuk 9 gaat in op een aantal praktische aspecten van het koppelen, die vaak voorkomen.

Het rapport wordt afgesloten met een literatuurlijst en drie appendices. In Appendix A wordt de koppelmethode van Fellegi en Sunter beschreven, die weliswaar in dit rapport geen belangrijke rol

speelt, maar historisch gezien van groot belang is. Appendix B gaat in op enkele zaken die met metrieken te maken hebben. Appendix C geeft een lijst met items die van belang zijn bij het kiezen van geschikte koppelsoftware.

1.6 Begrippen en definities

In tabel 1.1 staan de belangrijkste begrippen (inclusief hun synoniemen) die in dit rapport een rol spelen, met een korte uitleg van hun betekenis. Ook de in dit rapport voorkomende specifieke afkortingen en hun betekenis zijn in deze tabel opgenomen.

Tabel 1.1: Verklaring van enkele begrippen en afkortingen met betrekking tot het koppelen

Begrip	Omschrijving
Afstandsfunctie	Zie: Metriek
Atomaire eenheid	Zie: Enkelvoudige eenheid
BEID	Unieke bedrijfsidentificator (bij het CBS) van zogenaamde bedrijfseenheden in het Algemeen Bedrijvenregister (ABR) van 8 posities. De bedrijfseenheid vormt, samen met de Ondernemingengroep en in mindere mate de rechtspersoon, de belangrijkste statistische eenheid op basis waarvan de economische statistieken van het CBS worden samengesteld.
Bipartiete digraph	Een digraph $G = (V, \overline{E})$ met V de puntenset en A de verzameling gerichte kanten (pijlen), waarbij de puntenset V uit twee disjuncte stukken V_1 , V_2 bestaat. Iedere pijl a in $\overline{E}$ in zo'n graph heeft de eigenschap dat één van de punten van a in V_1 ligt en de andere in V_2 . In dit stuk hebben we te maken met een speciale deelklasse van bipartiete digraphen, namelijk die waarbij alle pijlen van V_1 naar V_2 lopen.
Bipartiete graph	Een graph $G = (V, E)$ met V de punten set en E de verzameling kanten, waarbij de puntenset V uit twee disjuncte stukken V_1 , V_2 bestaat. Iedere kant e in zo'n graph heeft de eigenschap dat één van de punten van e in V_1 ligt en de andere in V_2 .
Blocking variable	Een variabele waarmee men koppelingsbestanden partitioneert, met de bedoeling om de zoekruimte te verkleinen. Als de blocking variabele bijvoorbeeld een woongemeente is bij een koppelprobleem waar personen gekoppeld worden, dan betekent dit dat alleen personen woonachtig in dezelfde gemeente (op een bepaalde tijd) koppelbaar zijn.
BSN	Burger Service Nummer. Voorheen sofinummer genaamd.
Cut-off-waarde	Een waarde om de koppelgewichten (naar boven of naar beneden) te begrenzen. Hierdoor kan men bijvoorbeeld paren records die een te hoog koppelgewicht hebben uitsluiten als kandidaat-koppelingen, omdat ze onvoldoende op elkaar lijken. Of omgekeerd, door de cut-off-waarde te verhogen, kan men meer kandidaat-koppelingen zien te verkrijgen, omdat men ook minder sterk op elkaar lijkende records als koppelkandidaten beschouwt.
Deterministisch koppelen	Een koppeltechniek waarbij geen kansmodel wordt gebruikt. Bij joinen is dit het geval, dus bij koppelen op primaire sleutel. Toegepast in de context van koppelen met secundaire sleutels is dit begrip verwarrend en meestal ook niet van toepassing. Ook al gebruikt men een 'deterministisch koppelvoor-schrift', het is zeer wel mogelijk dat er koppelfouten worden gemaakt doordat er fouten en onregelmatigheden in de data voorkomen. Deze wijze van koppelen wordt dan gebruikt als tegenhanger van ' probabilistisch koppelen'. In dit stuk wordt dit begrip vermeden omdat het verwarrend is en tot misverstanden kan leiden.
Direct identifier	Zie: directe identificator
Directe identificator	Een variabele die gebruikt kan worden bij het identificeren van entiteiten. Dit omvat sleutelvariabelen (primary keys), maar ook variabelen als BSN, naam, adres, etc. die gebruikt kunnen worden om deze entiteiten recht-streeks te re-identificeren. Sommige directe identificatoren (zoals BSN) zijn

Begrip	Omschrijving
	geschikt als primaire sleutels. Andere (als naam, adres, etc.) zijn geschikt als secundaire sleutelvariabelen. Zie ook: indirecte identificator.
Dissimilarity measure	Een maat om de ongelijkheid van twee objecten of entiteiten uit te drukken. Lijkt enigszins op een metriek. Tegenovergestelde begrip: Similarity measure.
Drempelwaarde	Zie: cut-off-waarde
Enkelvoudige eenheid	Een eenheid die (voor het koppelprobleem in kwestie) niet is samengesteld uit eenheden van een lagere orde, ook wel enkelvoudige (of atomaire) eenheid genoemd. Een persoon is voor het CBS een enkelvoudige eenheid. Voor een arts kan het een samengestelde eenheid zijn, namelijk wanneer deze een persoon beschouwd als een samenstel van organen. Of een eenheid als enkelvoudig of samengesteld wordt beschouwd hangt van het koppelprobleem in kwestie af. Tegengesteld aan samengestelde eenheid.
ETL	Extract Transform Load. Een set operaties om een externe data-set geschikt te maken om (bij het CBS, zeg) verder bewerkt te worden. Deze operaties kunnen erop gericht zijn dataformaten om te zetten, nieuwe variabelen aan te passen, gebruikte coderingen om te zetten naar coderingen die het CBS hanteert, etc.
False negative match	Zie: Gemiste koppeling
False positive match	Zie: Miskoppeling
Fellegi-Sunter methode	Koppelmethode beschreven in Fellegi en Sunter (1969). Voor een korte bespreking van deze methode zie Appendix A.
Foreign key	Een sleutelwaarde, die wel in een record voorkomt maar niet gericht is om het record zelf te identificeren. Een foreign key bevindt zich derhalve buiten de sleutel van een dataset. Een foreign key is bedoeld om een koppeling te kunnen maken met een record in een andere dataset waarin bijvoorbeeld additionele gegevens op basis van die sleutel zijn opgenomen. Voorbeeld: in een record van een bedrijf, dat geïdentificeerd wordt door een Beid, is ook een unieke code, als foreign key, opgenomen van de regio waarin het bedrijf actief is. In een andere dataset is de code van de regio de primaire sleutel met additionele gegevens over de regio, zoals het aantal inwoners, de gemiddelde omzet van de bedrijven in die regio, de oppervlakte van de regio, e.d. In een record met persoonsgegevens, uniek geïdentificeerd door een BSN, kan men denken aan een verwijzing naar het bedrijf waar iemand werkt. Daarvoor kan bijvoorbeeld een code (bijvoorbeeld Beid) worden gebruikt. Een andere dataset, waar de Beid de sleutel is, bevat gegevens over het bedrijf waar de persoon werkt. Een foreign key is vaak een verwijzing naar een ander eenheidstype dan waar het record zelf betrekking op heeft, maar dat hoeft niet. Denk bijvoorbeeld aan gegevens van een werknemer met een verwijzing naar zijn baas. Beide zijn van het type persoon en beide zijn aan te duiden met een personeelsnummer.
Gemiste koppeling	Koppeling die ten onrechte niet gemaakt is.
Gewicht	Zie: koppelgewicht
Graad	Zij $G = (V, E)$ een graph. De graad van een punt v in V het aantal kanten e in E waarvoor geldt: $v \in e$.
Graadrestrictie	Beperking met betrekking tot de graad van een deel van de punten van een graph
Hamming-afstand	Afstand tussen twee records op een koppelsleutel, gemeten door het aantal variabelen te tellen waar de waarden verschillend zijn.
Incidentiematrix	0-1 matrix J die voor een graph $G = (V, E)$ aan geeft wat de relatie is tussen kanten in E en punten in V. Stel $\mid V \mid = n$, $\mid E \mid = m$ en J is de $m \times n$ matrix met $J(i, j) = 1$ als punt j op kant i ligt, en $J(i, j) = 0$ anders.
Indirecte identificator	Een variabele die gebruikt kan worden om (sommige) entiteiten in een populatie op te sporen, maar die geen directe identificator is. Voorbeelden zijn: woonplaats, beroep, leeftijd, geslacht. Indirecte identificatoren zijn kandidaten voor secundaire sleutels. Variabelen die geen directe of indirecte identificator zijn drukken bijvoorbeeld meningen, opinies, opvattingen e.d. uit. Dergelijke variabelen zijn niet geschikt als secundaire koppelsleutels. De scores van eenheden op dergelijke variabelen zijn in het algemeen niet publiek, en bovendien kunnen ze fluctueren in de tijd.
Indirect identifier	Zie indirecte identificator
Integer programming	Een speciaal geval van lineair programmeren, waarbij de variabelen die in het optimaliseringsmodel voorkomen integers zijn en geen reële getallen.

Begrip	Omschrijving
Joinen	Een vorm van koppelen die bij databases wordt toegepast en waarbij bijvoorbeeld op exacte gelijkheid van koppelsleutels wordt gekoppeld. (equi-join).
KK-digraph	Koppelkandidatendigraph (zie aldaar)
KK-graph	Koppelkandidatengraph (zie aldaar)
Koppelen	Het proces om gegevens (weergegeven in records) met betrekking tot eenheden en verspreid over twee bestanden, bij elkaar te brengen, op grond van (bijna) gemeenschappelijke kenmerken in de vorm van primaire of secundaire sleutelwaarden. Dit koppelen kan eenvoudig zijn, namelijk als gemeenschappelijke primaire sleutels aanwezig zijn in deze bestanden. Het kan ook een stuk moeilijker zijn, namelijk als er alleen secundaire sleutels aanwezig zijn, waarbij scores ook nog eens fouten kunnen bevatten, of wanneer deze variabelen niet helemaal identiek zijn.
Koppelfout van de 1e soort	Zie: Miskoppeling
Koppelfout van de 2e soort	Zie: Gemiste koppeling
Koppelgewicht	Voor een graph $G = (V, E)$ is een functie $w : E \rightarrow [0, \infty)$ een gewichtsfunctie, die dus met iedere kant van de G een niet-negatieve waarde G associeert. Bij het koppelen drukt dit gewicht uit hoe goed /slecht records koppelen. Het hangt af van de situatie of een hoger/lager koppelgewicht betekent dat koppelkandidaten beter/slechter bij elkaar passen.
Koppelgraph	Graph die het resultaat is van een koppeling. Het is een subgraph van de KK-graph
Koppelkandidatendigraph	Een bipartiete digraph die de mogelijke koppelingen tussen records uit twee bestanden representeert. De asymmetrie in de digraph kan bijvoorbeeld een gevolg zijn van de verschillende tijdstippen waar de koppelbestanden betrekking op hebben. De pijlen geven dan bijvoorbeeld een mogelijke ontwikkeling aan van een eenheid in het ene bestand in een eenheid uit het andere bestand. Bij de pijlen kunnen wel of geen koppelgewichten staan. Een koppelkandidatendigraph symboliseert een deel van de constraints die voor een koppelprobleem gelden. Afgekort KK-digraph
Koppelkandidatengraph	Een bipartiete graph die de mogelijke koppelingen tussen records uit twee bestanden representeert. Bij de kanten kunnen wel of geen koppelgewichten staan. Een koppelkandidatengraph symboliseert een deel van de constraints die voor een koppelprobleem gelden. Afgekort KK-graph
Koppelsleutel	Eén of meerdere sleutelvariabelen die in twee of meer te koppelen bestanden gebruikt worden, bijvoorbeeld om bij records uit het ene bestand records uit het andere bestand te zoeken. Als de koppelsleutel een primaire sleutelvariabele is zal koppelen op gelijkheid van de sleutel op zich weinig problemen opleveren. Als er echter een koppelsleutel wordt gebruikt die bestaat uit een aantal secundaire sleutelvariabelen dan zal het koppelen over het algemeen lastiger zijn vanwege fouten (of andere anomalieën) in scores op deze variabelen. Echter ook in primaire sleutels kunnen fouten voorkomen.
Lineaire programmering	In het Engels bekend als Linear Programming, afgekort LP. Dit is het gebied waarbij oplossingen gezocht worden voor problemen met lineaire doelfuncties die onder lineaire restricties dienen te worden geoptimaliseerd. De variabelen zijn hierbij reëelwaardig. Belangrijke subklassen worden gevormd door problemen waarbij alle, of sommige variabelen waarden aannemen in een eindige verzameling (bijv. {0,1}) of aftelbare deelverzameling (bijv. de natuurlijke getallen met 0). Dit laatste geval wordt wel geheeltallig programmeren of integer programming genoemd.
Matching	Engelse aanduiding voor koppelen; zie aldaar.
Metriek	Een metriek d op een verzameling X gedefinieerd is een functie $d : X \times X \rightarrow [0, \infty)$, dus een niet negatieve functie, met de volgende eigenschappen: 1. $d(x, y) = 0$ dan en slechts dan als $x = y$, 2. $d(x, y) = d(y, x)$ voor alle x, y in X (symmetrie), en 3. $d(x, z) \leq d(x, y) + d(y, z)$ voor alle x, y, z in X (driehoeksongelijkheid). Soms geldt in plaats van 3. een sterkere eigenschap: 4. $d(x, z) \leq \min\{d(x, y), d(y, z)\}$ Een niet negatieve functie d die aan 1, 2 en 4. voldoet heet een ultra-metriek.
Miskoppeling	Koppeling die ten onrechte gemaakt is.

Begrip	Omschrijving
Ontdubbelen	De records uit een bestand halen, op één na, die meerdere keren voorkomen, die allemaal op eenzelfde eenheid (in een bepaalde periode) betrekking hebben.
Primary key	Zie: primaire sleutel
Primaire sleutel	Primaire sleutel is (in databasetechnologie) de benaming voor een variabele of een combinatie van variabelen die voldoet aan volgende eisen : - de waarde van de variabele (of van de combinatie van variabelen) is uniek binnen de tabel (of dataset) en bepaalt dus eenduidig het record waarin hij voorkomt. - de variabele (of de combinatie van variabelen) is overal ingevuld en kan dus niet leeg zijn. De combinatie van variabelen is minimaal: door het laten vallen van één van de variabelen wordt de record niet langer meer eenduidig bepaald Wanneer gerelateerde tabellen verwijzen naar de tabel waarin de variabele (of combinatie) van variabelen voorkomen, wordt deze gebruikt om een relatie tussen tabellen tot stand te brengen. Voorbeelden zijn BSN en RIN-nummer voor personen, en BEID voor bedrijven. In de statistische beveiliging worden dergelijke variabelen ook wel directe identificatoren genoemd. Helaas worden daar variabelen als naam, adres, woonplaats, etc. ook directe identificatoren genoemd. Die zijn hier uitdrukkelijk niet bedoeld. Dergelijke variabelen worden in dit stuk secundaire sleutels genoemd.
Probabilistisch koppelen	Koppelen met als doel om informatie van dezelfde eenheden bij elkaar te zoeken, waarbij de scores op de koppelvariabelen niet per se hetzelfde hoeven te zijn. De verschillen kunnen diverse oorzaken hebben: 1. er zitten waarnemings- of verwerkingsfouten in de scores 2. de eenheden in beide bestanden zijn op verschillende tijdstippen waargenomen, of 3. koppelvariabelen in de verschillende files zijn niet exact hetzelfde gedefinieerd en hebben mogelijk andere domeinen.
Record linkage	Engelse aanduiding voor koppelen; zie aldaar.
Referentiële integriteit	In een relationele database is dit het uitgangspunt dat de interne consistentie tussen de verschillende tabellen binnen die database wordt gewaarborgd. Dat betekent dat er altijd een sleutel in een tabel bestaat als er in een sleutelveld, kan ook een foreign key zijn, in een andere tabel naar wordt verwezen. Database systemen waarborgen de consistentie en zorgen er voor dat een transactie die de consistentie doorbreekt niet kan worden doorgevoerd. Voorbeeld: er bestaat een tabel (1) met regiogegevens, geïdentificeerd door de postcode. In een andere tabel (2) wordt de postcode gebruikt om aan te geven in welke regio iemand woont. Referentiële integriteit zorgt ervoor dat de postcodes in tabel 2 altijd terug te vinden zijn in de tabel 1. Het kan niet zo zijn dat postcodes in tabel 1 worden verwijderd als deze nog voorkomen in tabel 2, dan wel als primaire, secundaire of foreign key.
Restant	Records die niet koppelbaar bleken bij een koppeling van twee bestanden. In sommige gevallen is het ongewenst dat er restanten overblijven en moeten ze worden 'weggewerkt' door extra koppelingen te realiseren.
RIN	Record Identificatie Nummer. Een primary key die door het CBS wordt gebruikt ter vervanging van ook buiten het CBS bekende sleutels als BSN. Dit gebeurt op basis van privacy overwegingen.
Samengestelde eenheid	Een eenheid die is samengesteld uit eenheden van een lagere orde. Een huishouden is een voorbeeld van een samengestelde eenheid; de personen die in een huishouden aanwezig zijn, zijn in dit geval eenheden van een lagere orde.
Samenhangende graph	Een graaf waarin alle punten verbonden zijn en dus één component vormen, heet een samenhangende graaf.
Samenhangscomponent (van een graph)	Binnen een graaf is een component een aantal knopen van de graaf die onderling alle verbonden zijn via een pad. Ook wel een (maximale) subgraph met deze punten plus de kanten van graph G met als eindpunten de punten uit de samenhangscomponent. Deze subgraph is een samenhangende graph. Hij is maximaal omdat er niet nog een punt aan kan worden toegevoegd zodanig dat de uitgebreide puntenset samenhangend is.
Secundaire sleutel	Een combinatie van variabelen die gebruikt kunnen worden bij de identificatie van eenheden, maar die niet bedoeld zijn als primaire sleutel. Vaak gaat het om (een combinatie van) variabelen zoals naam, adres, woonplaats, geboortedatum, beroep, opleiding, geslacht, e.d. Elk van deze variabelen zelf kan het record niet identificeren, maar de combinatie ervan kan gebuikt worden als proxy voor een primaire sleutel, mocht die

Begrip	Omschrijving
	ontbreken. In de statistische beveiliging worden dergelijke variabelen ook wel (indirect) identifiers genoemd.
Similarity measure	Maat die aangeeft hoezeer twee eenheden op elkaar lijken. Dit soort maten (of hun complementen: dissimilarity measures) worden ook in de multivariate analyse wel gebruikt, bijvoorbeeld om te clusteren.
Sleutel	Zie Primaire sleutel, Secundaire sleutel
Soundex-algoritme	Oorspronkelijk een fonetisch algoritme om namen te indexeren op basis van klank, uitgesproken in het Engels. Later is er een dergelijk algoritme ontwikkeld voor woorden in het Nederlands. Verbeteringen van het Soundex-algoritme voor het Engels zijn bijvoorbeeld Metaphone en Double Metaphone.
Surjectie	Een functie $f : X \rightarrow Y$ is een surjectie als voor iedere $y \in Y$ er een $x \in X$ is, zodanig dat $y = f(x)$. Zo'n f heet ook wel surjectief.
Statistisch koppelen	Koppelen van records met informatie van eenheden die niet per se dezelfde hoeven zijn, maar gelijksoortig. Qua intentie betreft deze methode een heel andere problematiek dan die in dit rapport worden besproken. Het betreft in feite een imputatiemethode. Om die reden wordt deze methode in dit rapport verder niet besproken..
Synthetisch koppelen	Zie: statistisch koppelen
Toegelaten koppelgraph	Een subgraph van een KK-graph die voldoet aan de criteria die gesteld worden aan de koppelgraph. Deze criteria hebben minimaal betrekking op de maximale graad van (een deel van) de punten (graadrestricties). In het Engels is 'toegelaten' feasible als in 'feasible solution'.
Type I error	Koppelfout van de 1e soort. Zie miskoppeling
Type II error	Koppelfout van de 2e soort. Zie gemiste koppeling
UWV	Uitvoeringsinstituut WerknemersVerzekeringen
Verrinnen	Records van een RIN nummer voorzien, die als een interne en beveiligde sleutel kan worden beschouwd. Gelijktijdig wordt de oorspronkelijke, externe sleutel (bijvoorbeeld een BSN/Sofi-nummer) uit het bestand verwijderd. In een aparte tabel zijn de externe sleutelnummers aan de RIN-nummers gekoppeld. Deze tabel wordt beveiligd opgeslagen

2. Overzicht van de koppelproblematiek

2.1 Wat is koppelen?

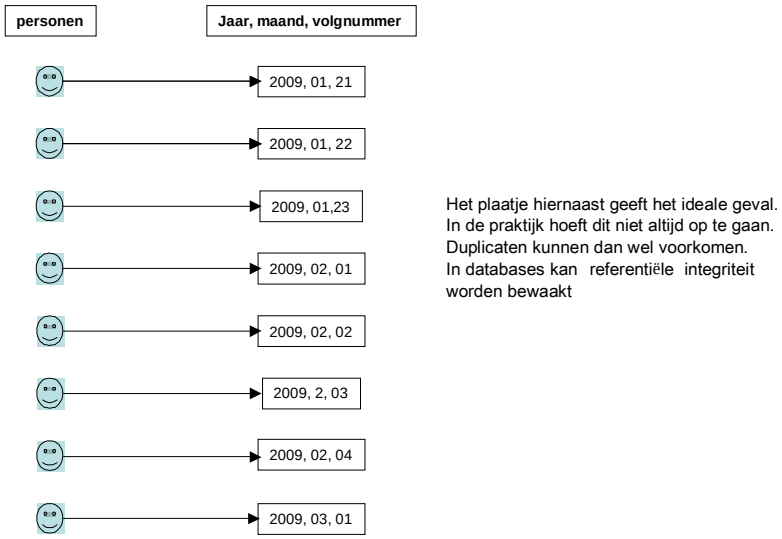
Koppelen is het bij elkaar brengen van de informatie van twee of meer records, waarvan gedacht wordt dat zij betrekking hebben op dezelfde eenheid zoals persoon, bedrijf of regio (zie Newcombe, 1988). Bij het koppelen worden gewoonlijk records, aanwezig in bijvoorbeeld twee verschillende files – koppelbestanden genoemd – bij elkaar gezocht op basis van verschillende criteria en randvoorwaarden.

Het koppelen gebeurt in *twee stappen*, namelijk:

1. eerst wordt nagegaan welke records *koppelkandidaten* zijn, en
2. vervolgens wordt uit alle mogelijke koppelkandidaten de *beste subset* gekozen, die aan bepaalde randvoorwaarden voldoet (bijvoorbeeld dat geen enkel record aan twee of meer records gekoppeld mag zijn).

In hoofdstuk 4 wordt nader ingegaan op beide stappen en de eisen die worden opgelegd aan toelaatbare oplossingen, waaruit de beste uiteindelijk dient te worden bepaald.

Figuur 2.1: Samengestelde primaire sleutel



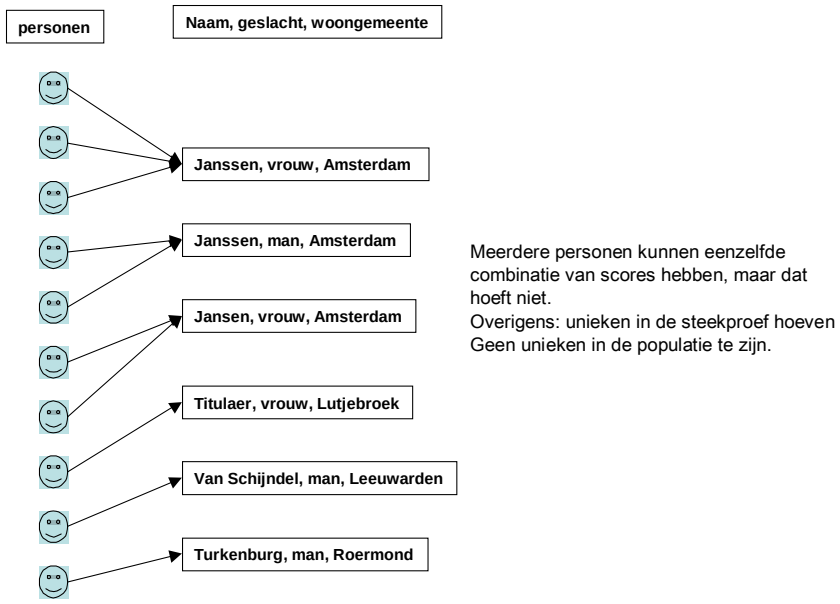
In dit stuk behandelen we twee *groepen* van koppel*methoden*.[2] In de eerste groep wordt op basis van een koppelcriterium in de vorm van een beslisregel, in de eerste fase van het koppelproces, nagegaan welke records koppelkandidaten zijn. Hiervoor wordt gewoonlijk gebruik gemaakt van een koppelsleutel bestaande uit een aantal variabelen die beide koppelbestanden gemeen hebben.

[2] We beschouwen het joinen dat in Hoofdstuk 5 wordt beschreven, strikt genomen, niet als een methode, maar als een procedure (in de terminologie van Van de Laar, 2008) omdat deze exact is. Dit is in tegenstelling tot methodes die worden gebruikt om benaderingen te vinden.

Het koppelcriterium kan dan bijvoorbeeld zijn: ‘exact gelijke scores op de koppelsleutel’. Dit criterium kan soms te streng zijn, omdat er ook fouten voorkomen in de scores van de koppelsleutels van de bestanden. Een verzwakking van dit koppelcriterium kan soelaas bieden. Bijvoorbeeld als de koppelsleutel uit meerdere koppelvariabelen bestaat: ‘exact gelijke scores op minstens m van de n koppelvariabelen’. Hierbij is n een gegeven parameter en m, met $0 < m < n$, een in te stellen parameter. In de tweede groep methoden wordt via een te berekenen koppelgewicht aangegeven in welke mate twee records matchen.

De beslissing voor het al dan niet koppelen van records (dus welke van de koppelkandidaten als koppelingen worden beschouwd) wordt over het algemeen door het koppelprogramma genomen. Als er interactief of handmatig wordt gekoppeld neemt een koppelspecialist deze beslissingen.

Figuur 2.2: Samengestelde secundaire sleutel



De te koppelen records kunnen geïdentificeerd worden door één variabele of een set van variabelen. Deze zogenaamde koppelvariabele(n) zijn meestal de identificerende sleutel van het record of de eenheid, de primaire (samengestelde) koppelsleutel. Zie figuur 2.1. Primaire koppelsleutels zijn eenduidig en, in theorie althans, kunnen er geen dubbele voorkomen. Dat hoeft echter niet altijd zo te zijn. Het kan ook gaan om een koppeling op basis van een andere variabele of variabelen in het record, de zogenaamde secundaire sleutels. Dergelijke sleutelvariabelen kunnen ook gebruikt worden om eenheden te identificeren, maar zijn minder hard en zijn niet ontworpen om eenheden eenduidig vast te leggen. Het is daarbij niet uitgesloten dat er dubbele voorkomen.

Niettemin kan een aantal van dergelijke secundaire sleutels vaak goed gebruikt worden om eenheden te identificeren en te koppelen.[3] Zie figuur 2.2.

In databases worden ook zogenaamde *foreign keys* gebruikt. Een foreign key identificeert het betreffende record zelf niet, maar is een verwijzing of koppeling naar een andere tabel, waarin de betreffende key wel als primaire sleutel voorkomt. Bijvoorbeeld om een record van een werknemer, geïdentificeerd door een personeelsnummer, te koppelen aan gegevens over het bedrijf, geïdentificeerd door een BEID, waar hij/zij werkt. In de tabel met werknemers is dan per werknemer record een BEID als foreign key beschikbaar die (uniek) koppelt aan de tabel met bedrijfsgegevens, waar de BEID de primaire key is. Voorwaarde daarbij is dat een foreign key waarde ook inderdaad bestaat, anders wordt verwezen naar een niet bestaande eenheid. Deze eigenschap wordt in databases wel aangeduid met 'referentiële integriteit'.

2.2 Wat maakt koppelen zo complex?

Op het eerste oog lijkt het koppelen van bestanden een eenvoudige opgave. In de praktijk is dat echter zelden het geval. De volgende oorzaken liggen onder meer ten grondslag aan het feit dat bestanden niet gemakkelijk één-op-één te koppelen zijn:

- de *kwaliteit en de structuur van de data* in de te koppelen bestanden. Het zal zelden zo zijn dat de aangeboden data, en dus ook die van de koppelvariabelen, zonder "ruis" is. Bij de verwerking kunnen bijvoorbeeld waarnemings- en verwerkingsfouten, zoals typefouten, optreden. Hierdoor kan het zijn dat records, die in werkelijkheid bij elkaar horen, niet koppelen of omgekeerd. Als het gaat om de structuur van de aangeboden data kan het bijvoorbeeld zijn dat de scores van de koppelvariabelen in beide records wel goed zijn, maar op een dusdanige manier zijn gepresenteerd dat het moeilijk is deze (geautomatiseerd) met elkaar te vergelijken. Dit alles maakt het stadium van pre-verwerking belangrijk. Daar kan zowel de kwaliteit als de structuur van de data, voor zover nodig bij het koppelen, worden aangepast en verbeterd;
- de *eenheden van te koppelen bestanden kunnen verschillen*, maar zijn wel uit elkaar af te leiden. Denk bijvoorbeeld aan een bestand met individuele personen enerzijds en een bestand met huishoudens anderzijds. Of een bestand met Bedrijfseenheden dat gekoppeld moet worden aan een bestand met Ondernemingengroepen. Hierbij dient gebruikt te worden gemaakt van een koppeltabel, waarin de relatie tussen beide eenheden is vastgelegd, of van een foreign key;
- het hanteren *van verschillende domeinen of classificatie-indelingen* bij de koppelvariabelen. Ook hier is het voor het koppelen wenselijk dat de domeinen of classificaties compatibel zijn, dat wil zeggen dat ze geconverteerd (kunnen) worden naar eenzelfde noemer (zonder te veel informatieverlies). Zie paragraaf 7.3.1.2 voor een verdere discussie van dit probleem;
- *de tijdsdimensie*. De koppelvariabelen of eenheden zijn dynamisch en zijn op verschillende momenten in de tijd waargenomen. Dat kan bijvoorbeeld gelden voor bedrijven. Tussen twee verschillende waarnemingen, die zijn opgeslagen in de twee verschillende bestanden, kan het bedrijf zijn gesplitst of juist gefuseerd, terwijl het bedrijf nog wel dezelfde identifier of

[3] Ook bij statistische beveiliging worden deze begrippen gebruikt. Daar zijn primaire sleutels in de regel niet aanwezig in beveiligde bestanden. De vraag is dan of de bestanden voldoende beveiligd zijn, lettend op de secundaire sleutels die in de bestanden aanwezig zijn.

koppelvariabele heeft. Bij het koppelen lijkt het om hetzelfde bedrijf te gaan, terwijl het in werkelijkheid kan gaan om een wezenlijk ander bedrijf. Een ander voorbeeld betreft een koppeling op woonadres dat bij verhuizing andere bewoners krijgt.

Deze verschillen in de te koppelen bestanden kunnen het koppelen complex maken en kunnen leiden tot de volgende twee typen fouten:

- records die weliswaar worden gekoppeld, maar in werkelijkheid niet tot dezelfde eenheid behoren (*miskoppeling, false positive match, type I error; koppelfout van de eerste soort*);
- records die niet worden gekoppeld, maar die in de werkelijkheid wel tot dezelfde eenheid behoren (*gemiste koppeling, false negative match, type II error; koppelfout van de tweede soort*).[4]

Dit soort fouten kan, ten slotte, ook nog geïntroduceerd worden door de keuzen die gemaakt worden bij het koppelproces zelf. Zo kan een foutieve of een te beperkte koppelsleutel worden gebruikt, kan de manier waarop de gewichten worden berekend verkeerd zijn, of kunnen de drempelwaarden of cut-off-waarden waartegen de gewichten worden afgezet, tot koppelfouten leiden.

Tabel 2.1: Mogelijke fouten bij het wel/niet koppelen van twee records

	De records horen bij eenzelfde eenheid	**De records horen niet bij eenzelfde eenheid**
De records zijn gekoppeld	- goede uitkomst - terecht gekoppeld	- miskoppeling - false positive match - type I error - ten onrechte gekoppeld
De records zijn niet gekoppeld	- gemiste koppeling - false negative match - type II error - ten onrechte niet gekoppeld	- goede uitkomst - terecht niet gekoppeld

2.3 Stappen in het koppelproces

Het koppelen op zich lijkt vaak relatief eenvoudig. In de praktijk pakt het toch vaak heel anders uit. Vooral de minder goede kwaliteit van de data en de onduidelijkheid daarover leiden tot allerlei problemen. Het koppelen kan niet los gezien worden van de voor- en nabewerkingsfase en het is daarom van belang stil te staan bij het koppel*proces*. Vier stadia kunnen daarbij worden onderkend (zie o.a. Gill, 2001; zie ook figuur 2.3):

[4] Dit gaat alleen over situaties met gelijke eenheden in beide koppelbestanden, dus waar effecten ten gevolge van de dynamiek van de populatie waar de eenheden toe behoren niet bestaan of verwaarloosbaar zijn.

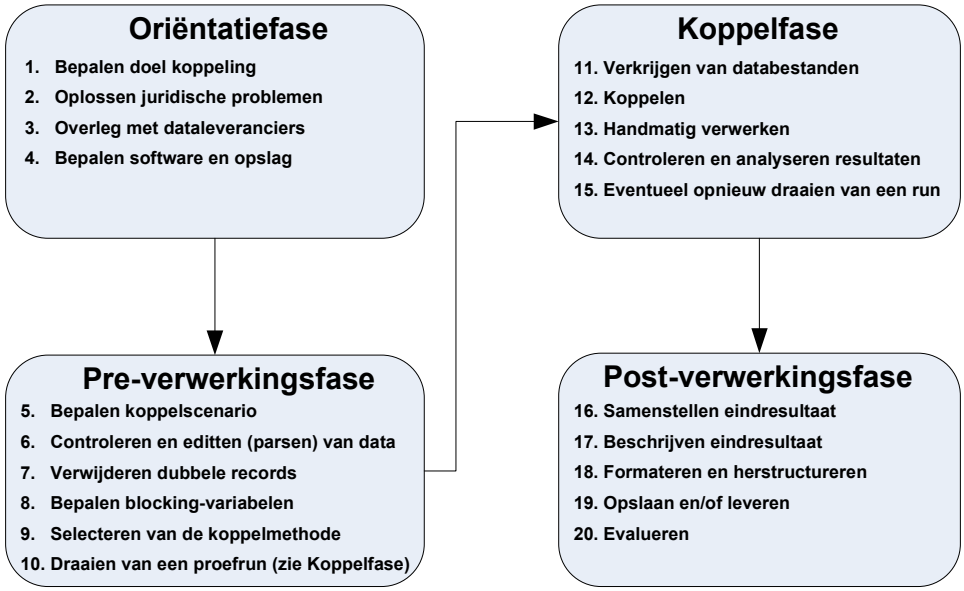
A. Oriëntatiefase:

1. Het vaststellen van het *doel van de koppeling*. Wat moet het resultaat zijn van de koppeling? Is het doel om zoveel mogelijk koppelingen te realiseren met een geringe mate van zekerheid of is men alleen geïnteresseerd in koppelingen die met een hoge mate van zekerheid zijn vastgesteld? Hoe erg is het om een koppeling te missen? Of vindt men het juist erg als een koppeling ten onrechte wordt gemaakt? Welke koppelvariabelen wil men gebruiken en wat is de kwaliteit van die koppelvariabelen? Dit soort aspecten bepalen voor een belangrijk deel uiteindelijk de andere stappen in het koppelproces;
2. Het oplossen van *juridische en ethische problemen*. Een eerste vraag is of er beperkingen zijn op het terrein van de privacy, eventueel vastgelegd in de wet. Daarvoor is bij het koppelen van personen bij het CBS bijvoorbeeld een RIN-nummer geïntroduceerd, dat voor de verwerking en het koppelen het originele BSN-nummer vervangt. Bij bedrijven dient men er zich rekenschap van te geven dat de resultaten van gekoppelde bestanden niet bij iedereen in de buitenwereld positieve reacties uitlokken. Doordat vaak gebruik wordt gemaakt van externe bestanden zijn ook afspraken nodig met de dataleverancier. Het is niet vanzelfsprekend dat de resultaten van gekoppelde bestanden vrij aan iedereen (bijvoorbeeld externe onderzoekers) ter beschikking kunnen worden gesteld. Dat geldt met name als het gaat om microdata. Een andere vraag is hoe de (fysieke) beveiliging van de data geregeld is.
3. Het overleggen met de (externe) *dataleveranciers* en het verkrijgen van de te koppelen bestanden. Hierbij gaat het om twee zaken. Ten eerste gaat het om de wijze waarop de bestanden worden geleverd en wat er wordt geleverd. Te denken valt aan informatie over de populatie en de betekenis van variabelen (inclusief het domein), het formaat en de structuur waarin de data worden geleverd. Periodieke levering van te koppelen bestanden moet uiteindelijk leiden tot goede onderlinge afspraken, die zijn vastgelegd in een SLA of SLL. Een tweede, even belangrijk, aspect is het verkrijgen van zoveel mogelijk informatie van de dataleverancier over (de kwaliteit van) de data in het bestand zelf. Daarbij gaat het niet alleen om informatie over de kwaliteit, maar ook om informatie over hoe de data tot stand is gekomen en verwerkt; hoe er is waargenomen; of er sprake is geweest van controles en zo ja welke dan; of er geen vreemde constructies zijn gebruikt, bijvoorbeeld door velden in een bestand te benutten voor doelen waarvoor ze niet zijn opgezet; hoe moet de kwaliteit van de data, en dan vooral van de koppelvariabelen, worden ingeschat. Kennis over dit soort zaken kan in een latere fase van het koppelproces heel veel werk schelen. Het ontwikkelen van deze benodigde kennis blijkt in de praktijk zeer arbeidsintensief te zijn. Leg de opgedane kennis dan ook vast. Dat is vooral van belang als er sprake is van veel mobiliteit bij medewerkers;
4. Het bepalen van de te gebruiken *software en opslag* van de (tussen)resultaten. Binnen het CBS wordt als koppelprogrammatuur gebruik gemaakt van TRILLIUM, eigen maatwerk en pakketten zoals MS Access en SPSS. De vraag is welke software het beste past bij het specifieke koppelprobleem. Voor de opslag van de (tussen)resultaten moet gekeken worden welke rustpunten kunnen worden benut: zijn lokale rustpunten beschikbaar of moeten die worden ontwikkeld, of kunnen de (tussen)resultaten in het Data Service Center (*DSC*), voor algemeen gebruik door anderen, worden opgeslagen?

B. Pre-verwerkingsfase:

5. Het bepalen van *het koppelscenario*. In de eerste plaats gaat het hierbij om de keuze welke koppelvariabelen te gebruiken, en om de vaststelling of inschatting van hun kwaliteit. Daarnaast kan gedacht worden aan de randvoorwaarden, die gesteld worden bij het koppelen, zoals de vraag of het om 1:1 koppelingen gaat of dat records in het ene bestand gekoppeld kunnen worden aan meer dan één record in het andere bestand.

Figuur 2.3: Fasen in het proces van het koppelen van bestanden



6. Het *controleren en editen* van de koppelvariabelen om de kwaliteit en de structuur te verbeteren. Het gebrek aan data zonder "ruis" is vaak het grootste obstakel bij het koppelen van gegevens. Het gaat om het converteren van de ruwe data naar meer gestandaardiseerde en consistente gegevens in een vorm die geschikt is om te koppelen. Te denken valt dan aan *het parsen* van variabelen, waarbij grotere, vaak free-format, strings of variabelen worden onderverdeeld in hun componenten en zoveel mogelijk worden gestandaardiseerd, zodat ze beter door de computer kunnen worden verwerkt en met elkaar kunnen worden vergeleken. Denk bijvoorbeeld aan het uit elkaar halen van de twee delen van de postcode, het numerieke deel en het alfanumerieke deel. Andere voorbeelden zijn het standaardiseren van adressen en namen, het "opvullen" van missings of simpele controles op spelfouten en waardebereik. Een ander aspect is het standaardiseren van classificaties over de te koppelen records heen. Voor al dit soort bewerkingen zijn een groot aantal verschillende technieken ontwikkeld. Zie o.a. Herzog e.a. (2007). Een laatste, niet vaak genoemde, mogelijkheid is het beter op elkaar afstemmen van de eenheden of koppelsleutels van de te koppelen bestanden. Een voorbeeld is de nieuwe Ondernemingengroep van de Eenhedenbase (*EHB*) bij de bedrijfseconomische statistieken. Deze is zo opgezet dat er een betere koppeling mogelijk is met de eenheden van de Belastingdienst;
7. Soms is het nodig om te onderzoeken of er *dubbele records* (duplicaten) in een bestand voorkomen, en deze te verwijderen indien dat het geval is ('ontdubbelen');

8. Het eventueel kiezen van de zogenaamde *blocking variabelen*. Twee te koppelen bestanden van bijvoorbeeld elk 1000 records leveren al 1.000.000 potentiële koppelkandidaten op. Om alle koppelkandidaten te controleren op een mogelijke koppeling is dan erg inefficiënt. Te koppelen bestanden zijn in de praktijk vaak nog vele malen groter en daarom wordt vaak gebruik gemaakt van zogenaamde blocking variabelen. Deze delen de te koppelen bestanden op zodat er twee of meer blokken ontstaan, waarbinnen records worden vergeleken. Is de kwaliteit van de gekozen blocking-variabele niet al te groot dan worden vaak meerdere runs gedraaid met verschillende variabelen als blocking-variabele. Men dient zich te realiseren dat het kiezen van de blocking variabele(n) niet evident is. Een "foute" keuze kan bijvoorbeeld leiden tot slechte eindresultaten;
9. Het selecteren van *koppelmethode*. Zie verder de hoofdstukken 5 tot en met 7;
10. Het eventueel uitvoeren, analyseren en beschrijven van een *proef-run* (zie daarvoor verder C. Koppelfase). Dit geldt vooral als een meer geavanceerde methode, bijvoorbeeld met koppelgewichten en cut-off-waarden, wordt gebruikt.

C. Koppelfase:

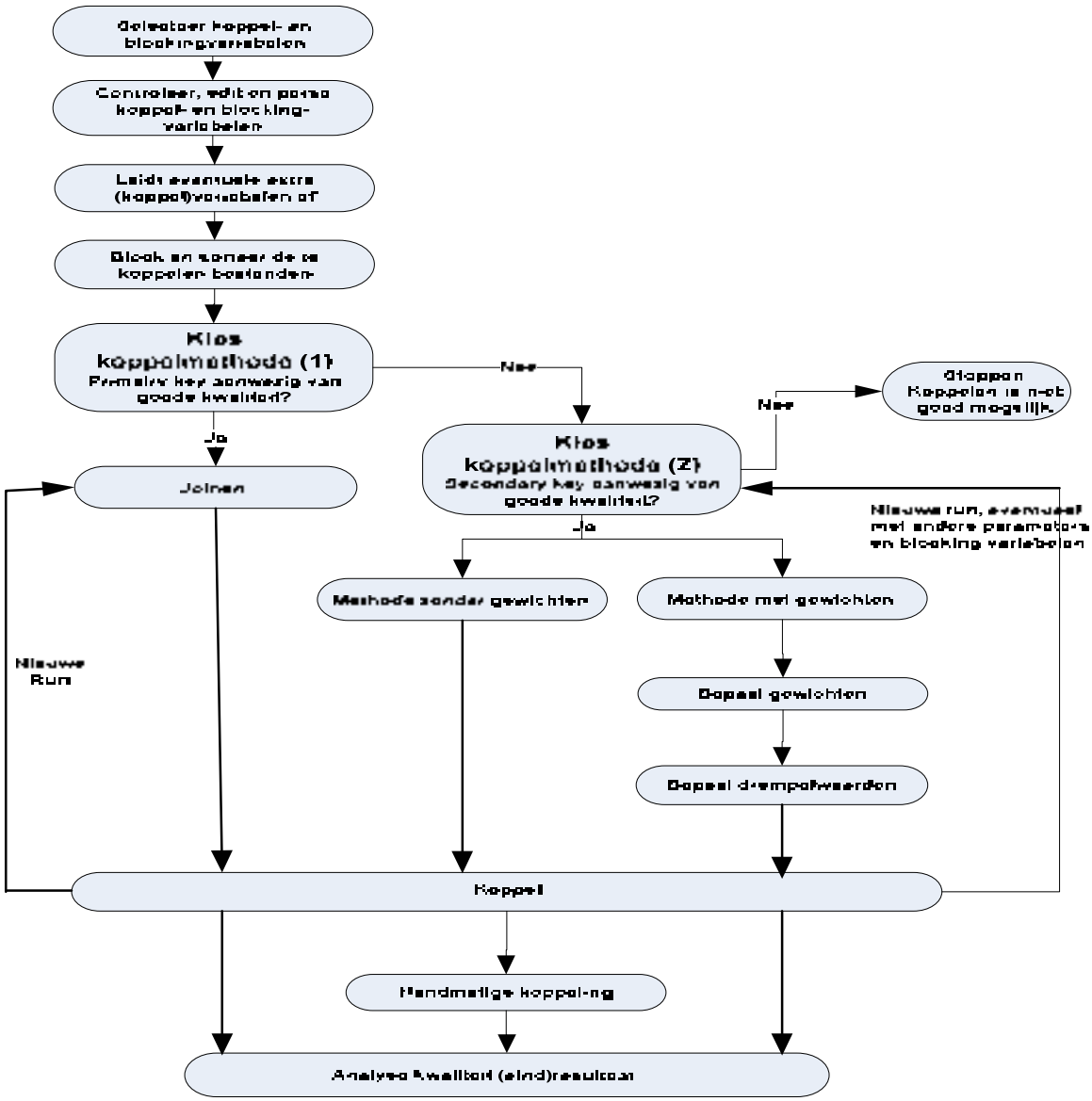
11. Het ophalen van de (eventueel eerder bewerkte) *databestanden;*
12. *Het koppelen zelf.* Daarbij kan onderscheid worden gemaakt naar:
    - het eventueel sorteren van de bestanden;
    - het eventueel bepalen van de *gewichten en cut-off-waarden*;
    - het bij elkaar brengen (*matchen*) van de potentieel te koppelen records op basis van de koppelsleutel (*de set van koppelkandidaten*), als eerste hoofdstap;
    - het *vergelijken van de verschillende koppelkandidaten en het besluiten of er sprake is van een "echte koppeling" of niet* of dat er sprake is van een twijfelgeval, als tweede hoofdstap;
    - het *opslaan in resultaatfiles.*
13. Het eventueel *handmatig verwerken* van de twijfelgevallen;
14. Het *controleren en analyseren* van het koppelresultaat en het bepalen van *kwaliteitsindicatoren* (type I en type II fouten). Een optie is om bijvoorbeeld een kleine steekproef te trekken uit het eindresultaat en handmatig te controleren of zij juist zijn of niet. Hiermee kan vervolgens een maat voor de kwaliteit worden berekend;
15. Het eventueel *opnieuw draaien van de run* met bijvoorbeeld andere blocking variabelen, gewichten en drempelwaarden of het minder streng toepassen van de voorwaarden bij de onderlinge vergelijking.

D. Post-verwerkingsfase:

16. *Samenstellen van het definitieve eindresultaat* (met bestanden van gekoppelde en niet gekoppelde eenheden);
17. Het *beschrijven van het eindresultaat,* onder meer met kwaliteitsindicatoren, het uitgevoerde proces en de gebruikte methoden (met parameters);
18. Het eventueel anders *formatteren en/of herstructureren van de eindresultaten* ten behoeve van de levering of de opslag;

19. Het *opslaan of leveren van de eindresultaten* aan de klant of aan de volgende stap in het proces;
20. Uitvoeren van een *evaluatie* van het verlopen proces zodat geleerd kan worden voor de volgende cyclus.

Figuur 2.4: Stappen in het koppelproces: pre-verwerkinsgfase en koppelfase



3. Graphen en metrieken (bij het koppelen)

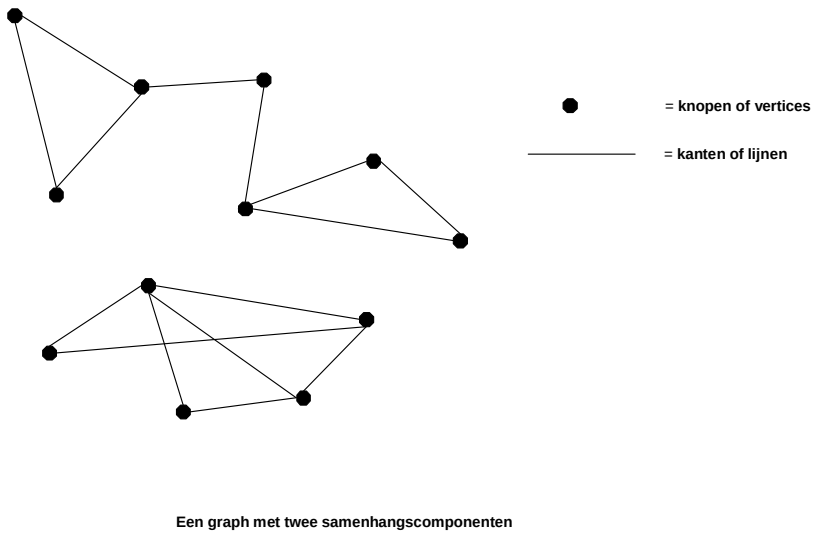
In dit rapport wordt voor het beschrijven van de theorie achter het koppelen en de indeling van de koppelmethoden gebruik gemaakt van de zogenaamde *graphentheorie*. Daarom wordt, voordat wordt ingegaan op de theorie en de koppelmethoden, in dit hoofdstuk kort stilgestaan bij wat de graphentheorie inhoudt. In het bijzonder worden enkele basisbegrippen verklaard die voor het navolgende van belang zijn.

Verder wordt in dit hoofdstuk ingegaan op een ander belangrijk begrip voor dit stuk, namelijk dat van *metrieken*. Een metriek wordt hier vooral gebruikt om de koppelgewichten te bepalen. Het onderwerp wordt hier geïntroduceerd. In Hoofstuk 7 wordt er dieper op ingegaan.

3.1 Graphen

Een graph G ($G = (V, E)$) bestaat uit een eindige verzameling punten V, ook wel knopen of vertices genoemd, waarvan sommige tweetallen verbonden zijn door lijnen (E), ook wel zijden, kanten of takken genoemd. In figuur 3.1 is een graph weergegeven.

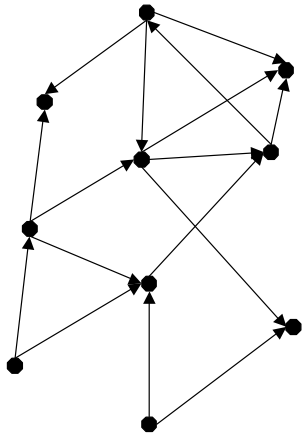
Figuur 3.1: Voorbeeld van een graph



Afhankelijk van de toepassing kunnen de lijnen gericht zijn, dan worden ze pijlen genoemd. In dat geval wordt gesproken van een gerichte graph of een *digraph*, afkorting van ‘directed graph’. Zie figuur 3.2. Ook kunnen gewichten aan de lijnen worden toegekend in de vorm van reële getallen.

Een graph met gewichten geassocieerd met punten of kanten noemt men een *gewogen graph*. In dit stuk zijn de gewichten geassocieerd met de kanten, en zij drukken een sterkte uit van de koppeling tussen twee records. Er zijn verschillende manieren om dergelijke gewichten te berekenen. In dit

Figuur 3.2: Voorbeeld van een digraph



stuk stellen deze gewichten koppelgewichten voor die de sterkte van potentiële koppelingen tussen records voorstellen. Men kan de gewichten ook zien als afstanden: hoe kleiner de afstand hoe meer de sleutels van de records op elkaar lijken.

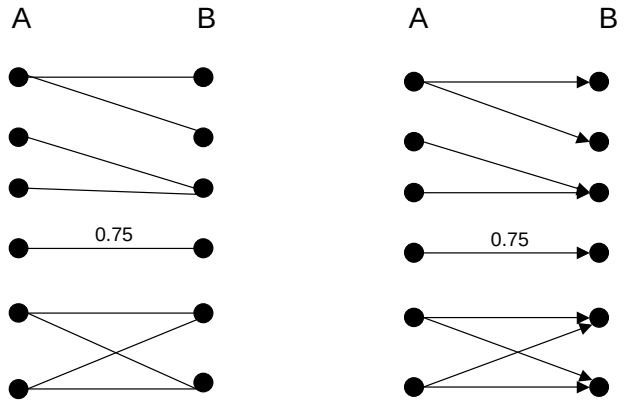
Een speciaal soort graph $G = (V, E)$ is de *bipartiete graph*. Zie figuur 3.3. Daarbij kan de verzameling knopen V worden verdeeld in twee disjuncte verzamelingen A en B. Er geldt dus: $V = A \cup B$ en $A \cap B = \phi$. De kanten verbinden uitsluitend knopen in A met knopen in B. Er zijn geen kanten binnen A en B zelf. Hierbij is het ook toegestaan dat één van beide verzamelingen leeg is, of zelfs allebei. De bipartiete graph is bij uitstek geschikt om (de theorie achter) het koppelen te illustreren.

Ten slotte wordt in dit stuk nog gesproken over de *KK-graph*, de *koppelkandidatengraph*. Het betreft hier een bipartiete graph die de mogelijke koppelingen tussen records uit twee bestanden representeert. Bij de kanten kunnen wel of geen koppelgewichten staan. Een koppelkandidaten-graph symboliseert een deel van de constraints die voor een koppelprobleem gelden.

Notatie:
Een graph G wordt genoteerd als een paar (V, E) waarin V de verzameling punten (knopen of vertices) voorstelt en E de verzameling lijnen of kanten. Iedere kant e in E is een verzameling $\{a, b\}$ met $a, b \in V$. Bij een gerichte graph, of digraph, is er sprake van een paar $(V, \overline{E})$ met V de verzameling punten en $\overline{E}$ de verzameling pijlen. Hierin is een pijl $\alpha \in \overline{E}$ een geordend paar

$(a,b) \in V \times V$. Met $|.|$ wordt de functie aangeduid die het aantal elementen van een verzameling weergeeft (eventueel, ∞ = oneindig). In dit stuk geldt voor alle graphen en digraphen dat ze eindig zijn, dat wil zeggen dat het aantal punten eindig is, dus $|V| < \infty$, en dus ook het aantal kanten (in graphen) of pijlen (in digraphen).

Figuur 3.3: Twee voorbeelden van een bipartiete graph



In beide voorbeelden vormen de punten twee disjuncte verzamelingen A en B. Kanten verbinden uitsluitend punten in A en B. Een kant kan voorzien zijn van een gewicht (getal >0).

De pijlen in het tweede voorbeeld zijn allemaal gericht van punten in A naar punten in B. Bij een algemene bipartiete digraph mogen de pijlen zowel van punten in A naar punten in B wijzen, als andersom.

Aantal samenhangscomponenten: 4.

Paden en samenhang in een graph:

Een *pad* in een graph is een opeenvolging van knopen op zo'n manier dat van elke knoop er een kant gaat naar de volgende knoop in de rij. Gegeven een graph $G = (V, E)$ met v en w als twee punten van G, dat wil zeggen $v, w \in V$. Een pad in G van v naar w is een rij $v_1,...,v_k$ van punten in G, zodanig dat:

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

Als er een pad van v naar w is in G dan ook een van w naar v (symmetrie). Als er een pad is in G van u naar v en van v naar w, dan ook van u naar w (transitiviteit). Hier zijn u,v,w punten in G. Voor ieder punt v in G is er – per definitie – een pad van v naar v (reflexiviteit). Met andere woorden, de relatie 'verbonden door een pad in een gegeven graph' is een *equivalentierelatie* op de verzameling punten van de graph, dus een binaire relatie die reflexief, symmetrisch en transitief is. Indien er slechts één equivalentieklasse is voor een graph G wordt G *samenhangend* genoemd. In dat geval zijn dus alle tweetallen punten door paden in G met elkaar te verbinden. Indien er twee of meer equivalentieklassen bestaan voor een graph, dan heet G *niet samenhangend*. In dat geval

correspondeert een equivalentieklasse van deze relatie met een *samenhangscomponent* van G, dit is een samenhangende subgraph van G. Zie bijvoorbeeld figuur 3.3.
Zij $G = (V, E)$ een graph, en zij v een punt in G, dus $v \in V$. De graad van v (in G) is het aantal kanten $e \in E$ waarvoor geldt dat $v \in e$, dus het aantal kanten in G waar v op ligt.

3.2 Metrieken

In dit stuk wordt ook gebruik gemaakt van *metrieken.* Een metriek is een functie, die de afstand tussen elk tweetal elementen van een verzameling definieert. Soms betreft het een functie die verwant is aan die van een metriek, maar die op enkele onderdelen van die van een metriek afwijkt. We krijgen dan gegeneraliseerde metrieken. Maar we beschouwen eerst metrieken.

We gaan uit van een verzameling X waarop een functie $d : X \times X \rightarrow [0, \infty)$ gedefinieerd is die aan een aantal condities voldoet.

1. $d(x, y) = 0$ dan en slechts dan als $x = y$,
2. $d(x, y) = d(y, x)$ voor alle x, y in X (symmetrie), en
3. $d(x, z) \leq d(x, y) + d(y, z)$ voor alle x, y, z in X (driehoeksongelijkheid).

Een niet-negatieve functie d die eigenschappen 1, 2 en 3 heeft, wordt een *metriek* genoemd. Soms geldt in plaats van eigenschap 3. een sterkere eigenschap:

4. $d(x, z) \leq \min\{d(x, y), d(y, z)\}$ voor alle x, y, z in X

Een niet-negatieve functie d die aan 1, 2, en 4. voldoet heet een *ultra-metriek*.

In de praktijk komen ook functies voor met eigenschappen die op een aantal onderdelen afwijken van die van een metriek. Zo is soms het bereik (co-domein) van d gelijk aan $[0, \infty]$, of geldt niet dat $d(x, y) = 0$ impliceert dat $x = y$, of geldt de symmetrie-eigenschap niet voor alle paren $x, y \in X$. Dit soort functies wordt algemeen wel aangeduid met gegeneraliseerde metrieken, en specifieke benamingen als pseudo-metriek, quasi-metriek, semi-metriek, hemi-metriek of (dis)similariteitsmaat worden gebruikt. In de statistiek worden ze onder andere bij clusteranalyse gebruikt (zie Mardia e.a., 1982).

Bij het koppelen en specifiek bij het vergelijken van koppelsleutels gaat het om het meten van de afstanden tussen de scores op de koppelsleutels, anders gezegd om de (on)vergelijkbaarheid te bepalen.

In het algemeen noteren we een metriek als d, of als $d(.,.)$, eventueel voorzien van een subscript (bijvoorbeeld d_H; zie beneden) waarbij de lengte van de koppelsleutel (het aantal koppelvariabelen waaruit deze bestaat) is weggelaten. De scores op een koppelsleutel noteren we als een vector $(\alpha_1, ..., \alpha_n)$ voor een koppelsleutel $(v_1, ..., v_n)$.

Van de metrieken die we gebruiken zijn er enkele zó speciaal dat ze apart worden aangeduid. We noemen hier de *Hamming-afstand,* genoteerd als d_H. Er geldt:

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

dus het aantal plaatsen waarop de vectoren α en β verschillende scores hebben. Merk op dat de Hamming-afstand (in principe) voor alle typen van variabelen te definiëren is.[5] Een voorbeeld: stel er zijn twee koppelsleutels van 4 alfanumerieke cijfers, respectievelijk "1034" en "1135". De Hamming-afstand is dan 2, omdat de cijfers verschillen op de 2 plekken, namelijk posities 2 en 4. Mat andere woorden: hoe kleiner de Hamming-afstand des te groter de vergelijkbaarheid van de koppelsleutels. De Hamming-afstand is gelijk aan het aantal "fouten" dat men in de ene sleutewaarde moet maken om de andere sleutelwaarde te verkrijgen.

Een andere metriek die we hier expliciet willen noemen is die van *Levenshtein*, genoteerd als d_L. Deze werkt op strings en telt het aantal elementaire operaties, zoals het weglaten, het veranderen of juist het toevoegen van karakters, nodig om de ene string in de andere te transformeren. In tegenstelling tot de Hamming-afstand, die een universele metriek kan worden genoemd, in de zin van: toepasbaar voor ieder type variabele, is de Levenshtein-afstand een metriek die specifiek voor het vergelijken van strings is ontworpen. Dit soort metrieken, specifiek toegesneden op een bepaald type variabele, zijn er meer, zoals in paragraaf 7.3.1 wordt getoond. Voorbeeld: de Levenshtein-afstand tussen de woorden "water" en "wetend" is 3: 1) water wordt weter (a vervangen door e), 2) weter wordt weten (r vervangen door n) en 3) weten wordt wetend (d wordt toegevoegd). Het voordeel van de Levenshtein-afstand, ten opzichte van de Hamming-afstand, is dat het sleutelwaarden van verschillende lengte kan verwerken.

Het volgende is een speciaal geval van een metriek voor een koppelsleutel bestaande uit meerdere variabelen. We zouden bijvoorbeeld een koppelsleutel kunnen hebben die bestaat uit n variabelen (alle secondaire sleutelvariabelen), allemaal van verschillend type, en waarbij de i^e variabele een metriek d_i heeft. We kunnen voor de hele koppelsleutel een metriek d definiëren door de metrieken van de *afzonderlijke variabelen van de sleutel gewogen op te tellen*, waarbij voor ieder gewicht w_i geldt $w_i > 0$, $i = 1,...,n$. We krijgen dan $d = \sum_i w_i d_i$. Overigens zijn de gewichten nodig om de afzonderlijke deelmetrieken op elkaar af te stemmen. Hierbij wordt gebruik gemaakt van het feit dat als δ een metriek is $a\delta$ dat ook is voor iedere $a > 0$. Door de gewichten te gebruiken kunnen we metrieken d_i op elkaar afstemmen. Overigens is het niet per se nodig om één metriek te gebruiken als we een koppelsleutel hebben bestaande uit meerdere variabelen. We zouden ook kunnen werken met de metrieken voor de afzonderlijke variabelen.

Soms is een indicatorvector nodig, die aangeeft op welke plaatsen α en β verschillen. δ zij een 0-1-indicator functie, die als volgt is gedefinieerd: $\delta(a,b) = 0$ als $a = b$ en $\delta(a,b) = 1$ als $a \neq b$, voor scores a, b voor een (koppel)variabele. Voor score vectors α, β zij

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

[5] Wat overigens niet wil zeggen dat het daarmee ook altijd een te verkiezen metriek zou zijn. Voor een alfanumerieke variabele bijvoorbeeld, zoals familienaam, zal men eerder een metriek kiezen die gradueel onderscheid maakt tussen verschillende namen. Zo wijkt 'Jansen' slechts één letter af van 'Janssen', maar van 'Boog' zes letters. De Hamming-afstand registreert slechts dat beide namen verschillend zijn.

Voor de methode van Fellegi en Sunter (1969) (zie Appendix A) speelt deze indicatorvector een centrale rol. Merk op dat $d_H(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} d(a_i, b_i)$.

4. Theorie van het koppelen

4.1 Inleiding

Stel dat we koppelbestanden A en B hebben, waarvoor geldt dat ze informatie bevatten die betrekking heeft op tijdstippen die niet al te ver uit elkaar liggen. Ten aanzien van de koppelbaarheid van deze bestanden doen zich de volgende mogelijkheden voor (zie ook figuur 2.4):

1. Er is een gemeenschappelijke en unieke primaire koppelsleutel die zowel in bestand A als in bestand B aanwezig is. Er zijn vervolgens twee mogelijkheden:
   a. De scores op de variabelen in de koppelsleutel zijn van voldoende kwaliteit.
   b. De scores op de variabelen in de koppelsleutel zijn van onvoldoende kwaliteit.
2. Er is geen (goede) gemeenschappelijke en unieke primaire koppelsleutel in beide bestanden aanwezig. Er zijn wel bepaalde variabelen gemeenschappelijk in beide bestanden aanwezig die als secundaire koppelsleutel kunnen dienen. Ook in dit geval zijn er twee mogelijkheden:
   a. De scores op deze gemeenschappelijke secundaire koppelsleutel zijn van voldoende kwaliteit.
   b. De scores op deze gemeenschappelijke secundaire koppelsleutel zijn van onvoldoende kwaliteit.

Het is duidelijk dat dit een opsomming is van typen van koppelproblemen, geordend van gemakkelijk (geval 1a) naar moeilijk of zelfs ondoenlijk (geval 2b). De lastigste gevallen van koppelen zijn de gevallen die onder 1b of onder 2a vallen. Daar ligt de koppelproblematiek waar in dit stuk de meeste aandacht aan zal worden besteed. Situatie 1a wordt voor de volledigheid ook behandeld (hoofdstuk 5) maar hier spelen geen methodologische problemen. In de terminologie van Van de Laar (2008) betreft dit een procedure en geen methode.[6]

Koppelcriteria en randvoorwaarden:
Toepassing van een *koppelcriterium* levert records op die mogelijk te koppelen zijn, de zogenaamde koppelkandidaten. Deze koppelkandidaten worden eerst vastgesteld, in geval koppelmethoden worden gebruikt, dat wil zeggen in situaties met secundaire sleutels en fouten of afwijkingen in de data. Als men bijvoorbeeld een metriek gebruikt om de afstand (of de mate van overeenkomst) tussen twee records te meten, dan geeft het koppelcriterium aan bij welke afstanden (bijvoorbeeld cut-off- of drempelwaarden) men moet stoppen twee records nog als koppelkandidaten te gebruiken. Stel men heeft vijf secundaire koppelvariabelen als samengestelde sleutel. Het koppelcriterium zou dan kunnen zijn dat records die op minimaal drie van de vijf koppelvariabelen een gelijke score hebben als koppelkandidaten beschouwd moeten worden en de rest niet.

[6] Het verschil is dat een methode een benadering betreft, en een procedure niet. Zo heeft men een ophoging nodig om via populatieschattingen benaderingen te krijgen voor populatieaantallen. Zo'n ophoging is gebaseerd op één van de vele ophoogmethoden die bekend zijn, en die ieder geschikt zijn voor specifieke situaties. Ophogingen leveren schattingen op van populatiegrootheden. Bij een koppeling van twee bestanden op basis van een harde sleutel (een primaire sleutel) is geen sprake van een benadering. (Zie Hoofdstuk 5) De koppelmethoden gebaseerd op secundaire sleutels (zie Hoofdstukken 6 en 7) zijn wel benaderingen.

Voor het koppelen van records in twee koppelbestanden hebben we niet alleen variabelen nodig in beide bestanden waarmee we de koppeling kunnen uitvoeren. Daarnaast dienen we de *randvoorwaarden* te weten waaronder gekoppeld moet worden.

Veelal wordt een koppeling zodanig uitgevoerd dat geen enkel record in beide koppelbestanden aan meer dan één record uit het andere bestand gekoppeld mag worden (1:1 koppelingen), waarbij het tevens mogelijk is dat records niet gekoppeld worden. Echter er zijn ook situaties dat andere randvoorwaarden gelden. Bijvoorbeeld dat ieder record uit het ene bestand, zeg A, gekoppeld moet zijn aan minstens één record uit het andere bestand, B, terwijl ieder record uit B met ten hoogste één record uit A gekoppeld mag zijn (1:n koppelingen). Dit vereist wel dat bestand B evenveel of meer records heeft dan bestand A. Deze situatie kan zich voordoen als de eenheden in A een deelverzameling vormen van de eenheden in B. In geval we bijvoorbeeld rekening moeten houden met splitsingen en fusies van eenheden, zijn de randvoorwaarden weer anders. In dat geval moet worden toegestaan dat meerdere eenheden uit bestand A met één of zelfs meerdere eenheden uit B gekoppeld kunnen worden, en omgekeerd (m:n koppelingen). Het is belangrijk om te weten voor een koppelprobleem onder welke randvoorwaarden een koppeling precies plaatsvindt.

Methode:
Om tot een koppelmethode te komen moet bekend zijn wat voor soort koppeling men wil uitvoeren, zonder koppelgewichten of juist mét zulke gewichten. In het eerste geval tellen de potentiële koppelingen die gevonden zijn allemaal even zwaar; in het tweede geval is dat niet zo, maar kan door middel van gewichten worden aangegeven hoe sterk een kandidaat-koppeling is.[7] Het berekenen van deze koppelgewichten kan op diverse manieren gebeuren. Men kan gebruik maken van metrieken, (dis)similariteitsmaten, kansmodellen etc. In paragraaf 7.3.1 wordt nader ingegaan op het berekenen van deze gewichten.

Doelfunctie:
Voor een bepaalde klasse van koppelproblemen (met koppelgewichten) wordt gebruikt gemaakt van een doelfunctie die moet worden geoptimaliseerd (geminimaliseerd) onder randvoorwaarden. In de doelfunctie worden de verschillende koppelkandidaten van een gewicht voorzien, het koppelgewicht. Dit koppelgewicht wordt gebruikt om de sterkte van potentiële koppelingen te kunnen nuanceren. Zoals hierboven is aangegeven kan men koppelgewichten op verschillende manieren berekenen.

Specifieke situaties:
Het kan zijn dat het op het eerste gezicht lijkt of er geen goede (secundaire) koppelsleutel beschikbaar is, namelijk als sommige van de variabelen van deze sleutels niet precies hetzelfde domein hebben. Voorbeeld: in het ene bestand komt een leeftijdsvariabele voor met als domein de leeftijd in jaren (zeg de verzameling $\{0,1,2,...,120\}$) terwijl het andere bestand een leeftijdsindeling in vijfjaarsklassen kent (zeg de verzameling $\{0-4,5-9,...,100+\}$. Ook in

[7] Formeel zijn de methoden waarbij geen koppelgewichten worden gebruikt een speciaal geval van methoden waarbij dat wel het geval is. Immers men kan gewichten associëren met alle kandidaat-koppelingen die allemaal dezelfde waarde hebben, bijvoorbeeld 1. Omdat de oplossingsmethoden anders zijn voor beide typen koppelmethoden, maken we het onderscheid hier wel (net als in de literatuur van de combinatorische optimalisering).

dergelijke situaties zijn deze variabelen als koppelvariabelen te gebruiken. De technieken die dan gebruikt kunnen worden zijn niet veel anders dan in het geval van situaties 1b of 2a.

Een andere specifieke situatie betreft het feit dat niet wordt voldaan aan de voorwaarde dat de scores in beide bestanden op ongeveer hetzelfde tijdstip betrekking dienen te hebben, dus met (ongeveer) gelijke referentietijden. Het kan zijn dat de tijdstippen[8], waarop de gegevens in de bestanden betrekking hebben, zo ver uit elkaar liggen dat er verschillen in de scores van dezelfde eenheden kunnen optreden louter omdat er sprake is van dynamiek in de populatie, waardoor nieuwe eenheden kunnen instromen in de populatie ('geboorte'), of juist uitstromen ('sterfte'), of van eigenschap veranderen (bijvoorbeeld een jaartje ouder worden, huwen of scheiden, etc.). Daarnaast komt het voor dat de eenheden zelf kunnen veranderen. Dit is bijvoorbeeld mogelijk bij samengestelde eenheden zoals bedrijven of huishoudens, die kunnen splitsen of fuseren met andere eenheden.

Koppelen wordt meestal gebruikt om records uit 2 bestanden 1:1 te matchen. Dit wil zeggen dat in de definitieve matching als twee records koppelen, r_A uit A en r_B uit B, er geen records s uit A zijn en t uit B, zodanig dat r_A aan t koppelt of r_B aan s. Het is echter zeer wel mogelijk dat afzonderlijke records uit A aan meerdere records uit B kunnen worden gekoppeld, of omgekeerd, afzonderlijke records uit B aan meerdere in A. Te denken valt hierbij aan koppelsituaties waarbij tussen de peiltijd waarop A en waarop B is verzameld een zodanig verschil zit dat de effecten van de dynamiek zichtbaar worden. Het kan dan zijn dat een bedrijf uit bestand A is gesplitst in meerdere bedrijven die in bestand B zijn vertegenwoordigd. Of omgekeerd dat een bedrijf in B is ontstaan door fusie van meerdere bedrijven in A. Het hoeft hierbij trouwens niet per se om samengestelde eenheden te gaan als bedrijven. Het kan ook om personen gaan. Zo zou een persoon die in bestand A 32 jaar is in bestand B 32 of 33 jaar oud kunnen zijn. Als op secundaire sleutelwaarden twee personen voorkomen in B die dezelfde scores hebben op alle koppelvariabelen, alleen een verschillende score op de variabele leeftijd, namelijk 32 jaar en 33 jaar, dan zijn beide personen in B koppelkandidaten voor het genoemde record in A (verondersteld dat er verder geen identificerende informatie in A en B aanwezig is). Met bepaalde kansen kunnen de beide records uit B aan het record in A worden gekoppeld. Hierbij kan de kans gebruikt worden als koppelgewicht, of beter gezegd de reciproke kans. Dit is dan een voorbeeld van een koppelingsmodel dat gebruik maakt van koppelingsgewichten. Dat is niet per se nodig. Er zijn ook koppelingsmodellen die zonder deze gewichten werken.

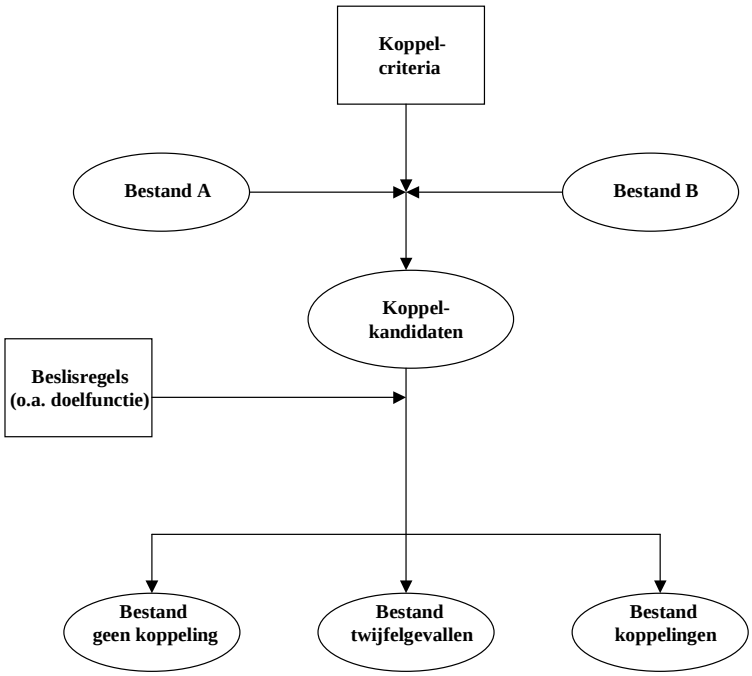
Verder wordt in dit stuk niet alleen gekeken naar koppelingen van gelijke eenheden. In de economische statistieken is vaak sprake van samengestelde eenheden, die kunnen splitsen, maar ook kunnen samengaan met andere, soortgelijke eenheden, tot een nieuwe eenheid. Zie ook paragraaf 9.6. De relatie tussen twee eenheden in verschillende te koppelen bestanden hoeft dan niet per se te zijn die van gelijkheid van eenheden, maar bijvoorbeeld van 'is voortgekomen uit' (bij een splitsing) of omgekeerd 'is onderdeel geworden van' (bij een fusie).

[8] Daarbij gaat het om het tijdstip waarop de data betrekking hebben. Echter ook al gaat het om hetzelfde tijdstip waarop de gegevens betrekking hebben, er kan ook sprake zijn van een groot tijdverschil in het moment van registratie. Ook in dat geval kunnen genoemde koppelproblemen optreden.

Ten slotte, is het bij het koppelen van belang te weten of alle records van het ene bestand gekoppeld moeten worden aan records uit het tweede bestand (of omgekeerd, of beide). En ook of men graag zo veel mogelijk records koppelt met een bijbehorend groot risico, of omgekeerd, dat men kiest voor een hoge kwaliteit van de gerealiseerde koppelingen, en op de koop toeneemt dat sommige koppelingen worden gemist.

We hebben nu alle ingrediënten van een koppelmethode besproken.[9] In het vervolg van dit hoofdstuk gaan we een en ander nader uitdiepen en toelichten. Dit alles ter voorbereiding op de uitgebreide besprekingen in hoofdstukken 5, 6 en 7.

Figuur 4.1: belangrijkste ingrediënten van het koppelen



4.2 (Keuze tussen) de methoden van koppelen

Koppelen heeft dus te maken met het in verband brengen van gegevens uit verschillende bestanden. Daarvoor wordt een koppelcriterium gebruikt en eventueel ook een doelfunctie en randvoorwaarden waaraan een toegestane koppeling moet voldoen. Daarnaast kan ook nog gekeken worden naar de intentie van dit bij elkaar brengen van gegevens, en in het bijzonder om welke populatie-eenheden het dan gaat. Als het om dezelfde eenheden gaat betreft het een koppeltechniek waar dit stuk over gaat. Als het er echter om soortgelijke, maar niet per se dezelfde, eenheden gaat dan is er sprake van *statistisch koppelen*, ook wel aangeduid als *synthetisch koppelen.* Zoals aangegeven, valt deze klasse van technieken buiten de scope van dit stuk, omdat deze methoden meer thuis horen (zowel qua intentie als qua uitvoering) bij imputatiemethoden.

[9] Bedoeld zijn vooral de lastigere koppelmodellen 1b en 2a. De situatie in 1a is triviaal en valt hier buiten.

In dit stuk bekijken we drie wijzen van koppelen:

- Koppelen op een (samengestelde) primaire sleutel ("joinen");
- Koppelen op een (samengestelde) secundaire sleutel, zonder gebruik te maken van koppelgewichten;
- Koppelen op een (samengestelde) secundaire sleutel, met gebruikmaking van koppelgewichten.

Koppelen op een primaire sleutel is eigenlijk de ideale manier van koppelen, omdat hier de eenheden een eenduidige en unieke identificatie kennen. In theorie kunnen er geen dubbele voorkomen. In de praktijk zullen er echter toch foutjes insluipen. Zo kan het zijn dat er in één bestand (ten onrechte) toch dubbele records voorkomen.[10] In dat geval moet er eerst ontdubbeld worden. Mochten er veel duplicaten zijn dan hebben we eigenlijk te maken met een koppelsleutel die niet erg betrouwbaar is. In dat geval zou men beter kunnen kijken naar alternatieven, in de vorm van secundaire sleutels. Die moeten dan wel voorhanden zijn in de koppelbestanden.

Indien er geen primaire sleutel gemeenschappelijk aanwezig is in beide koppelbestanden, maar wel een secundaire koppelsleutel, kan deze gebruikt worden om records te koppelen. We moeten hier echter van uitgaan dat er meer koppelfouten mogelijk zijn: koppelingen kunnen worden gemaakt terwijl ze niet terecht zijn, of gemist, terwijl ook dat niet de bedoeling is. We beschouwen voor deze situatie twee mogelijkheden: er worden geen of juist wel koppelgewichten gebruikt om de sterkte van een mogelijke koppeling weer te geven.

Welke methode men wil kiezen hangt sterk af van de situatie. Te noemen zijn:

- *de kwaliteit van de koppelsleutels*. Is de kwaliteit van de koppelvariabelen goed en sterk identificerend dan zal men eerder kiezen voor het "joinen" (als dus een goede unieke primaire sleutel aanwezig is) of een methode zonder gewichten;
- *de eenduidigheid en vergelijkbaarheid van de koppelsleutels*. Zijn de koppelsleutels onderling sterk discriminerend dan komt het gebruik van methoden zonder gewichten eerder in beeld dan een methode met gewichten;
- *de gewenste kwaliteit van de koppelingen*. De methoden zonder koppelgewichten hebben, grofweg, betere performance en boeten in op kwaliteit, terwijl bij modellen met koppelgewichten juist een betere kwaliteit en een geringere performance te verwachten is;
- *beschikbare hardware en software*. In veel koppelsoftware is het gebruik van gewichten vaak niet (goed) mogelijk en, zoals gezegd, zullen "joinen" en een methode zonder gewichten beter performen dan een methode met gewichten;
- *tijd, beschikbare capaciteit en kennis*. Is deze beperkt dan zal men eerder kiezen voor het gebruiken van "joinen" (als een goede primaire sleutel aanwezig is) of een methode zonder

[10] Overigens is het meerdere keren voorkomen van records met eenzelfde primaire sleutel niet per definitie fout. Denk bijvoorbeeld aan een banenbestand, of een voertuigenbestand. Een persoon kan best meerdere banen hebben, of meerdere voertuigen bezitten. Het gaat dan om foreign keys. In het persoonsbestand dient iedere persoon een eenduidig persoonsnummer te hebben (BSN) Er zijn echter ook situaties dat de afzonderlijke records in een bestand geacht worden betrekking te hebben op verschillende eenheden, bijvoorbeeld personen. In zo'n geval moeten alle records in het bestand een verschillende sleutelwaarde hebben. Het voorkomen van twee records met eenzelfde sleutelwaarde is dan een fout.

gewichten, omdat deze eenvoudiger uit te voeren zijn dan een methode met gewichten en minder kennis vraagt. Aangetekend zij dat zo'n keuze gezien de kwaliteit en vergelijkbaarheid van de koppelsleutels niet optimaal hoeft te zijn. Een methode met gewichten vergt (initieel) veel tijd. Zo moeten niet alleen de gewichten worden bepaald (inclusief de wijze waarop), maar ook de juiste hoogte van de cut-off-waarden of drempelwaarden, waarboven of waaronder koppelkandidaten nog wel worden gezien als echte matches of juist niet. Om deze gewichten en waarden goed te kunnen vaststellen en daar enig gevoel voor te krijgen zijn meerdere runs van de koppeling noodzakelijk. Voordeel van een methode met gewichten is dat men – door de drempelwaarden te verhogen of te verlagen - kan spelen met de hoeveelheid koppelingen die als twijfelachtig worden gezien en dus met de capaciteit die nodig is voor handmatige verwerking.

4.3 Koppelmodellen op basis van graphen

In dit hoofdstuk beschrijven we koppelen vanuit een theoretisch gezichtspunt en laten we zien hoe het modelleren van het koppelprobleem in zijn werk gaat, namelijk: de probleemformulering en de oplossing. Daarvoor moeten in de praktijk ook bepaalde parameters worden ingevuld, toegesneden op de concrete koppelsituatie. Een voorbeeld hiervan vormen de zogenaamde koppelgewichten.

Centraal in de bespreking van de koppelmethoden in dit hoofdstuk vormt de zogenaamde KK-graph[11] bij een koppeling van twee bestanden. Dit is een bipartiete graph, waarbij de ene set punten de records in het eerste koppelbestand, zeg A, voorstellen en de andere set punten de records in het andere koppelbestand, zeg B. De kanten of lijnen in de KK-graph stellen potentiële koppelingen voor.

We maken in dit stuk, naast het "joinen", onderscheid tussen twee groepen methoden, namelijk: methoden zonder gewichten en methoden met gewichten. De eerste groep methoden werkt met een KK-graph waarbij alle kanten als gelijkwaardig worden beschouwd. Bij de tweede groep methoden is dat niet zo. De verschillen tussen de kanten in de KK-graph worden uitgedrukt door middel van koppelgewichten. In dit hoofdstuk wordt kort ingegaan op de verschillende methoden die bestaan om tot koppelgewichten te komen. Een belangrijke deelklasse van koppelmethoden die tot de tweede groep behoort, vormen de probabilistische koppelmethoden. Dat we dergelijke technieken nodig hebben heeft te maken met onzekerheid bij het koppelen door fouten of onregelmatigheden in de data, gebruik van enigszins verschillende koppelvariabelen of dynamiek in de populaties, waardoor kenmerken van eenheden kunnen veranderen na verloop van tijd, maar niet altijd op een eenduidige manier. Een bedrijf kan na een zekere periode nog bestaan zoals voorheen, of failliet zijn gegaan of zijn gefuseerd, of zijn overgenomen, en dit allemaal met verschillende kansen.

Bij het koppelen in de praktijk zal meestal sprake zijn van grote KK-graphen, waarbij er tevens relatief weinig gevallen zijn van koppelparen waarbij één van de punten in meerdere mogelijke koppelparen aanwezig is. Het probleem (en het werk) bij het koppelen komt daarom vooral neer op het berekenen van kandidaat-koppelingen en de eventueel bijbehorende gewichten en niet zozeer in het maken van keuze uit alternatieve kandidaat-koppelingen, want die komen naar verwachting relatief weinig voor. Bovendien zal het daar vaak gaan om relatief kleine keuzeproblemen die los

[11] Zie Hoofdstuk 3 voor meer informatie over graphen en metrieken.

van elkaar opgelost kunnen worden. Ieder van deze keuzeproblemen correspondeert met een samenhangscomponent van de desbetreffende KK-graph.

Een ander punt is dat in de statistiek gevallen met alternatieve koppelingen als twijfelgevallen worden beschouwd, die dienen te worden voorgelegd aan een koppeldeskundige om ze op te lossen. Dat is vaak echter niet nodig, en zou een programma die beslissingen kunnen nemen. Dat scheelt niet alleen een hoop werk, het kan bovendien leiden tot betere (automatische) procesdocumentatie. Alleen échte probleemgevallen kunnen dan aan een koppeldeskundige worden voorgelegd. Dit mogen er echter niet meer dan een handjevol zijn.

Voordat we ons concentreren op de koppelingsproblemen bij bipartiete (di)graphen, kijken we eerst naar het koppelingsprobleem bij willekeurige (di)graphen.

4.4 Koppelproblemen in graphen

Het koppelen kan worden beschreven met behulp van een speciaal soort graphen, namelijk bipartiete graphen. Echter in een algemene graph $G = (V, E)$ kan men ook over matching spreken. Het eenvoudigste geval is 1-matching (of gewoon matching genaamd). Het doel hierbij is om een subset $F \subseteq E$ te kiezen zó dat ieder punt $v \in V$ op ten hoogste één kant van F ligt. Dit laatste kan ook zo geformuleerd worden dat voor $G(F) = (V, F)$ de graad van ieder punt v in $G(F)$ ten hoogste 1 is. Iedere graph heeft altijd één 1-matching, namelijk met $F = \phi$, dus de graph die uit de punten van G bestaat en geen kanten heeft. De kunst is nu om bij een gegeven graph $G = (V, E)$ een maximale 1-matching $G(F) = (V, F)$ te vinden, dus waarbij het aantal kanten $|F|$ maximaal is. Dit is een *ongewogen matchingsprobleem.*

1-Matchings zijn te generaliseren tot h-matchings, waarbij h een $|V|$-vector is van integers, waarbij h_i een bovengrens is voor het aantal kanten waar punt $i \in V$ op ligt. Bij h-matchings kan een extra eis worden opgelegd namelijk dat iedere kant niet meer dan één keer gekozen mag worden. Dit wordt 0-1 h-matching genoemd. Een andere mogelijkheid is dat een kant meerdere keren kan worden gekozen. Dat wordt aangeduid als integer h-matching. In dit stuk maken we alleen gebruik van 0-1 h-matching.

Aan de kanten, de koppelkandidaten, kunnen ook gewichten worden toegekend. Als w_e een gewicht is voor $e \in E$ dan zij $w(E') = \sum_{e \in E'} w_e$ het gewicht voor $E' \subseteq E$. Het *gewogen h-matchingsprobleem* is om een h-matching te vinden van maximaal gewicht. Het ongewogen 1-matchingsprobleem is een speciaal geval hiervan, omdat hier $w_e = 1$ voor iedere $e \in E$.

Een formulering van gewogen 0-1 (1-)matching als integer programmingsprobleem is als volgt:

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

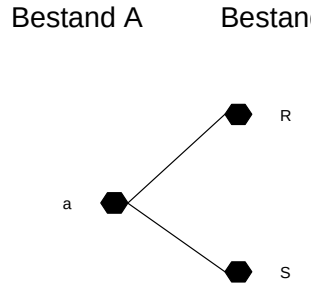
waar $A = (a_{ij})$ de incidentiematrix met $a_{ij} = 1$ als punt j op kant i ligt en $a_{ij} = 0$ als dat niet zo is.

Verder is $b = (\overbrace{1,...,1}^{m})'$, met m het aantal kanten, ofwel $m = |E|$. De vector b bestaat uit louter 1-en, omdat we met een 1-matching te maken hebben. Verder is $n = |V|$, het aantal punten in G. $x_e = 1$ betekent dat kant e in de matching zit en $x_e = 0$ dat dat niet zo is.

Matchingsproblemen kunnen worden geformuleerd als *optimaliseringsproblemen*. Of een maximum of minimum moet worden gevonden onder restricties laten we in dit stuk afhangen van het matchingsprobleem is kwestie. Soms is het natuurlijker een maximum te gebruiken, en in andere gevalleen een minimum. Uiteraard zijn beide soorten problemen eenvoudig in elkaar om te zetten door een minteken voor de doelfunctie te zetten. Voor een uitgebreide behandeling van matchen in de combinatorische optimalisering, zie bijvoorbeeld Nemhauser en Wolsey (1988, hoofdstuk III.2) of Papadimitriou en Steiglitz (1998, hoofdstukken 10 en 11).

Een mogelijke aanpak voor het oplossen van de koppelproblematiek is als volgt. Eerst elimineren we de records in beide bestanden die met geen enkel record koppelen. Dit levert een eerste reductie van de KK-graph op. Vervolgens kan men dan nog de matches zonder alternatieven oplossen. Dit levert een verdere reductie op van het probleem. Als er nog iets overblijft zijn dat kandidaat-matches met alternatieven. Deze kunnen worden opgelost volgens bovenstaande methode (de combinatorische optimalisering). Hier heeft men in de praktijk in de regel te maken met meerdere kleinere en onderling onafhankelijke problemen waar uit alternatieve koppelingen gekozen moet worden. Als dit er veel zijn is het het beste dat een programma deze keuzes maakt (en de beslissingen vastlegt in een logfile). Lastiger gevallen moeten daarna door koppeldeskundigen worden bekeken en beoordeeld; gemakkelijkere gevallen wellicht alleen op steekproefbasis. Lastig en eenvoudig is hier op basis van de grootte van het op te lossen keuzeproblemen: hoe groter hoe lastiger.

Figuur 4.2: twee mogelijke, gelijkwaardige koppelingen



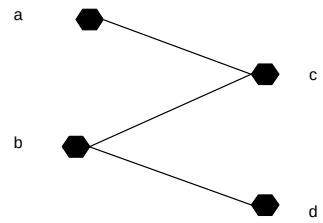
Koppeling {a,R} en {a,S} zijn koppelkandidaten. Bij 1:1 koppeling moet voor één van beide koppelingen worden gekozen en de andere koppeling vervalt dan.

4.5 Werkwijzen

Stel we hebben twee bestanden die gekoppeld moeten worden. Eerst wordt vastgesteld welke koppelsleutel moet worden gebruikt. Naast de koppelsleutel zouden nog andere grootheden en parameters gebruikt kunnen worden, zoals gewichten en cut-off-waarden. Het doel is vervolgens om te komen tot een criterium om records uit bestand A te koppelen aan records uit bestand B. Indien dit criterium wordt toegepast op de records in bestanden A en B levert dat paren records op die ieder aan het koppelcriterium voldoen en die dus koppelbaar zijn. Dergelijke informatie is weer te geven in de vorm van een KK-graph.

Afhankelijk van de gekozen koppelmethode, kan het zijn dat de sterkte van de koppelingen nog moet worden uitgedrukt, in de vorm van koppelgewichten. Hiervoor moet dan eerst een geschikte karakterisering worden gevonden om die te bepalen. Soms kan het koppelcriterium daarbij worden gebruikt, waarbij bijvoorbeeld de mate van afwijking van het ideaal kan worden gekwantificeerd.

Figuur 4.3: Twee mogelijke koppelingen



Mogelijke koppelingen bij 1:1 koppeling: 1. {a,c} en {b,d} en 2. {b,c}

Indien de KK-graph is samengesteld, al of niet met koppelgewichten, is het daarna van belang om een geschikte doelfunctie te formuleren, die aangeeft wat voor soort koppelingen men zoekt. Verder moeten er criteria worden geformuleerd waaraan de oplossing moet voldoen. Dit soort criteria hebben veelal te maken met de maximale graad voor alle punten van de koppelgraph. In veel gevallen moet bijvoorbeeld gelden dat de graad maximaal 1 mag zijn. Dit zou bijvoorbeeld kunnen gelden als de te koppelen eenheden personen zijn. Er zijn echter ook situaties waarbij een 1:n koppeling mogelijk is. Dit is bijvoorbeeld het geval bij splitsingen van bedrijven, waarbij één bedrijf in twee of meer onderdelen wordt gesplitst die ieder afzonderlijk doorgaan. Ook een

koppeling n:1 is denkbaar, bijvoorbeeld – om de bedrijvensfeer te blijven – als 2 (of meer) bedrijven fuseren. Zelfs n:m koppelingen zijn denkbaar, bijvoorbeeld als n bedrijven samengaan en uit dit totaal m nieuwe bedrijven worden opgericht. Op dergelijke voorbeelden gaan we verderop dieper in. Alleen zij nog opgemerkt dat de interpretatie van de koppeling in deze gevallen vaak anders is. Hier worden geen gelijke eenheden gekoppeld, maar eenheden die voort zijn gekomen uit andere eenheden.

In figuur 4.3 zijn er bijvoorbeeld 2 koppelkandidaten beschikbaar. Uitgaande van het criterium dat slechts 1:1 koppelingen toegestaan zijn, is geen keuze te maken, tenzij men een keuze kan maken op basis van bijvoorbeeld berekende gewichten. We vinden dus twee mogelijke koppelingen: {a,c} en {b,d}. In principe is er echter a priori niets op tegen dat {b,c} als koppeling wordt verkozen. Dat is zeker zo als er met gewichten zou worden gewerkt en het gewicht bij kant {b,c} groter zou zijn dan dat van de gewichten geassocieerd met de kanten {a,c} en {b,d} samen.

Als {b,c} als koppeling wordt verkozen dan worden de mogelijke matches{a,c} en {b,d} dus uiteindelijk niet gemaakt, bij de voorwaarde dat 1:1-koppelingen moeten worden gemaakt.

4.5.1 Koppelen op primaire sleutel

Het koppelen op een unieke primaire sleutel waarbij ook nog eens de scores (de sleutelwaarden) in beide koppelbestanden van hoge kwaliteit zijn, is in zekere zin triviaal. Dit soort koppelingen worden in databasepakketten standaard uitgevoerd. De operatie heet daar *join*, of preciezer, *equi-join*.

Koppelen op unieke primaire sleutel heeft alleen zin als de sleutelwaarden ook redelijk betrouwbaar zijn, en er gekoppeld kan worden op exacte gelijkheid van sleutelwaarden.[12] Het heeft geen zin om, als de waarden niet betrouwbaar zouden zijn, te werken met een metriek en koppelkandidaten te zoeken die ‘in de buurt’ van sofi-nummers liggen. Naburigheid van sofi-nummers heeft geen enkele relatie met naburigheid van personen in de normale betekenis van het woord.

4.5.2 Koppelen op secundaire sleutels, zonder koppelgewichten

Het doel van de eerste stap, de modelformulering, is om een koppelprobleem te specificeren in de vorm van een optimaliseringsprobleem. Om dit te kunnen doen moeten een aantal zaken worden gekozen door de degene die verantwoordelijk is voor de uit te voeren koppeling. Op basis van deze specificaties / keuzes moeten vervolgens zaken worden afgeleid uit de koppelbestanden. We gaan op ieder van deze onderdelen in.

Voor koppelproblemen zonder koppelgewichten zijn er de volgende zaken die moeten worden gespecificeerd:

1. De koppelsleutel, bestaande uit de variabelen uit de beide koppelbestanden die men wenst te koppelen.

[12] Nodig is dan dat de referentietijden voor de beide koppelbestanden niet te ver uit elkaar liggen. Wat ‘ver’ is wordt bepaald door de dynamiek in de desbetreffende populatie en de koppelfouten die men wenst te accepteren.

2. Het koppelcriterium dat men wenst te gebruiken om koppelkandidaten te kunnen berekenen. Dit koppelcriterium toegepast op de koppelbestanden levert een KK-graph op.
3. Bij grote koppelbestanden kan het nodig zijn om met een stratificatie te werken die de zoekruimte voor het vinden van koppelkandidaten beperkt. Hiervoor worden zogenaamde blockvariabelen gebruikt. Eén (of meer) blokvariabelen kunnen worden gebruikt om zo'n stratificatie te berekenen.
4. Graadrestricties die gelden voor de koppelgraph. Dit betekent dus dat koppelingen 1:1, 1:n, m:1 of m:n moeten zijn. Het kan ook zijn dat de m of n van boven begrensd moet zijn. Dat één record aan meerdere records uit het andere bestand te koppelen is kan voorkomen als de peildata van de koppelbestanden verschillen.

Nadat een koppelmodel zonder koppelgewichten is gespecificeerd als optimaliseringsprobleem, is het zaak dit probleem op te lossen. Oplossingsmethoden voor dit soort modellen bespreken we in hoofdstuk 6.

4.5.3 Koppelen op secundaire sleutels, met koppelgewichten

Voor koppelproblemen met koppelgewichten zijn er, in aanvulling op de 4 items die in paragraaf 4.5.2 worden genoemd, nog een aantal andere zaken te specificeren, namelijk:

5. Wijze van berekening van de koppelgewichten. Bij het berekenen van de KK-graph kunnen tevens deze gewichten worden berekend.
6. Cut-off [13] waarde die aan geeft welke koppelingen men nog wenst te accepteren als acceptabel. Dit is een drempelwaarde die ervoor zorgt koppelgewichten die te klein zijn, dat wil zeggen beneden een door de koppelaar aan te geven benedengrens, dienen niet beschouwd te worden als kandidaat-koppelingen. Met deze cut-off-waarden kan men het risico beïnvloeden koppelingen te missen, maar tevens, en dat is de keerzijde, om koppelingen ten onrechte te maken.
7. Specificatie van de doelfunctie, bij het gebruik van koppelgewichten. In de regel is dit eenvoudigweg de som van de koppelgewichten van de kanten in een toegelaten koppelgraph.

Oplossingsmethoden voor dit soort modellen bespreken we in hoofdstuk 7.

Een koppelprobleem – afgezien van een koppelprobleem op een primaire sleutel - willen we in dit stuk definiëren als een optimaliseringsprobleem. Hierbij moet, op basis van criteria die moeten gelden voor de oplossing, een subgraph van de KK-graph wordt bepaald die de doelfunctie in het probleem optimaliseert.

[13] Hierbij kan ook tegelijkertijd worden gewerkt met een boven- en ondergrens. Alle koppelingen met gewichten boven de bovengrens worden als ware koppelingen gezien. Alle koppelingen beneden de benedengrens worden als ware mismatches gezien. Koppelingen die in het gebied liggen tussen deze twee grenzen betreffen de twijfelgevallen en worden aangeboden aan de koppelspecialist. Door te spelen met de boven- en ondergrens kan de omvang van het aantal twijfelgevallen worden beperkt of uitgebreid.

5. Koppelen op primaire sleutel

5.1 Korte beschrijving

Koppelen op een primaire sleutel is de eenvoudigste manier van koppelen. In beide koppelbestanden komt dezelfde unieke primaire sleutel voor die als koppelsleutel wordt gebruikt. Aanname is dat de kwaliteit van de primaire sleutel voldoende hoog is; anders kan deze koppelwijze niet goed worden toegepast. Deze vorm van koppelen wordt erg vaak gebruikt, met name ook omdat weinig koppelkennis vereist is en de methode eenvoudig en niet erg bewerkelijk is. Daarnaast wordt deze methode ondersteund door veel softwarepakketten, lopend vanaf Excel en Access tot meer geavanceerde database- en koppelpakketten.

Uitgangspunt: een koppeling vindt dan en slechts dan plaats als een record uit het ene koppelbestand precies dezelfde sleutelwaarde heeft als die van een ander record uit het tweede koppelbestand. Dit soort koppelingen vindt standaard in databases plaats, omdat database-managementpakketten daar functionaliteit voor in huis hebben. In database termen is er sprake van een operatie genaamd ‘join’, of ‘equi-join’.

In de zin van Van de Laar (2008) betreft het een procedure en geen methode, omdat er geen sprake is van een benadering. Bij de andere vormen van koppelen – op secundaire sleutels – is dat juist wel het geval. Daar gaat het dan om methoden.

Ter verduidelijking: als het gaat om een unieke primaire sleutel dan gaat het veelal om een sleutel die bestaat uit één sleutelvariabele, zoals Burger Servicenummer of Bedrijfsidentificatie. Dat hoeft echter niet zo te zijn. Een primaire sleutel kan best uit meerdere componenten of variabelen bestaan, bijvoorbeeld een sleutel voor huishoudens met een volgnummer voor de leden in een huishouden. Als sleutel voor de personen in huishoudens is deze combinatie uniek. Dat de sleutel mogelijk is opgebouwd uit meerdere componenten of variabelen is in feite niet van belang. Bij een primaire sleutel gaat het erom dat het record uniek wordt geïdentificeerd. Er kunnen, in theorie althans, geen dubbele voorkomen. Bij secundaire sleutels, die in de hoofdstukken 6 en 7 een centrale rol spelen, is veelal sprake van meerdere, echt verschillende variabelen die kunnen worden gebruikt om eenheden te koppelen. Het is in dat geval niet uitgesloten dat er dubbele voorkomen. Vandaar dat daar gesproken wordt van secundaire koppelsleutels of sleutelvariabelen.

5.2 Toepasbaarheid

Aanname bij deze methode is dat de gehanteerde koppelsleutels in beide bestanden van goede kwaliteit zijn. Dat is echter niet altijd het geval, of de kwaliteit van de koppelsleutels is onbekend, niet goed onderzocht. Toch wordt deze vorm van koppelen erg vaak gebruikt. De methode is eenvoudig en weinig bewerkelijk. Zij vereist ook weinig kennis.

De toepasbaarheid van deze wijze van koppelen is zeer ruim, namelijk in die gevallen waarbij unieke primaire koppelsleutels van goede kwaliteit aanwezig zijn.

Mocht de kwaliteit tegenvallen, maar men beschikt over een volledige lijst primaire sleutels met informatie van de eenheden die daarbij horen, dan zou men toch nog iets kunnen doen. Stel men heeft bijvoorbeeld als primaire sleutel BSN, en men beschikt over een complete lijst met BSN’s met (enige) informatie over de desbetreffende personen. Indien men een BSN-nummer tegenkomt dat niet lijkt te kloppen dan zou men ‘in de buurt’ kunnen kijken van dit nummer in de lijst. Het

idee daarbij is dat bij het kopiëren van zo'n nummer een fout is gemaakt, bijvoorbeeld dat twee cijfers zijn omgewisseld, of een 5 door een 6 is vervangen (of omgekeerd) of een 7 door een 1 (of omgekeerd) etc. Door bijvoorbeeld alle BSN's op te zoeken met een Levenshtein-afstand 1 of 2 (zie paragraaf 3.2) tot de gegeven BSN, en door de bijbehorende persoonskenmerken te vergelijken met de gegevens in het betreffende bestand of register, zou men wellicht toch een correct BSN kunnen vinden met bijbehorende persoonskenmerken. Hierbij gaat het in feite om een methode die thuishoort in de hoofdstukken 6 en 7.

5.3 Uitgebreide beschrijving

Gezien de eenvoud van deze methode valt er niet zoveel meer over te zeggen. Uitgangspunt is dat een koppeling plaatsvindt dan en slechts dan als een record uit het ene koppelbestand precies dezelfde sleutelwaarde heeft als die van een ander record uit het tweede koppelbestand. Dat kan betekenen dat er sprake is van 1:1, 1:n of n:1 koppelingen. Als de kwaliteit van de primaire koppelsleutel onvoldoende (bekend) is, bestaat het gevaar dat koppelingen worden gelegd, die geen echte koppelingen zijn (miskoppelingen), en omgekeerd (gemiste koppelingen).

5.4 Voorbeelden

We geven de volgende voorbeelden van koppelsituaties waar op primaire sleutel wordt gekoppeld:

- de koppeling van bedrijven uit twee statistieken, die allebei op het Algemeen Bedrijvenregister (ABR) zijn gebaseerd. In beide bestanden wordt de eenheid, het bedrijf, geïdentificeerd door een Bedrijfsidentificatienummer van 8 cijfers (BEID). Het BEID is dan de primaire sleutel waarop wordt gekoppeld. Komen BEIDs in beide bestanden overeen dan wordt er gekoppeld; komen ze niet overeen dan worden ze niet gekoppeld. Er wordt bijvoorbeeld geen rekening meegehouden met het feit dat er tijdens het verwerkingsproces van de individuele statistieken fouten in de BEIDs kunnen zijn geslopen. Deze controle is vaak ook moeilijk omdat in veel gevallen geen secundaire sleutels, zoals naam en adres, meer aanwezig zijn;
- een variant op het eerste voorbeeld is dat gegevens van de Belastingdienst gekoppeld worden aan de BEIDs van het ABR. In het ene bestand van het ABR is de primaire sleutel BEID aanwezig. In het bestand van de Belastingdienst is de Fiscale eenheid (FE) als primaire sleutel aanwezig. Om beide bestanden aan elkaar te koppelen is er een "relatie- of koppeltabel" aanwezig, waarbij is aangegeven welke FE's bij welke BEIDs behoren. Op dezelfde wijze als in het eerste voorbeeld, alleen met een extra tussenstap, worden beide bestanden aan elkaar gekoppeld. De kans op foute koppelingen is hier wel groter omdat er niet alleen fouten kunnen zijn geslopen in de FE's of BEIDs, maar ook in de registratie van de relatie tussen FE's en BEIDs;
- bij persoonsgegevens is vaak het Burgerservicenummer (BSN) als primaire sleutel in het bestand aanwezig. In dat soort gevallen is een eenvoudige koppeling te maken op basis van de BSN in beide bestanden. Een voorbeeld is het koppelen van loon- en werkgelegenheidsgegevens van respectievelijk de Belastingdienst en de Polisadministratie (van het UWV);
- koppeling op basis van een zogenaamde foreign key, bijvoorbeeld de SBI-codering of grootteklasse-codering in een record van een BEID. Kengetallen of gemiddelden uit een bestand op basis van de SBI of grootteklasse, maar dan als primaire sleutel, kunnen aan het

record worden gekoppeld van het bestand met BEIDs. Dit gebeurt vaak bij het gaafmaken of imputeren.

Opgemerkt zij dat een externe primaire sleutel voor koppelwerkzaamheden op het CBS kan worden vervangen door een uitsluitend binnen het CBS betekenisvolle, en dus te gebruiken, sleutel. Dit proces wordt wel aangeduid als verrinnen, dat wil zeggen het toekennen van een RIN-nummer. De bedoeling hiervan is databeveiliging, om te voorkomen dat allerlei andere informatie aan een bestand gekoppeld kan worden op basis van een externe primaire sleutel, zoals een BSN.

5.5 Kwaliteitsindicatoren

De kwaliteit van dit soort koppelingen is geheel afhankelijk van de kwaliteit van de primaire sleutel. Vaak wordt te gemakkelijk voorondersteld dat deze van voldoende kwaliteit is. Het is ook mogelijk dat alleen de primaire sleutel in het bestand aanwezig is en er geen secundaire sleutels beschikbaar zijn. Dat is bijvoorbeeld het geval als bestanden zijn verRINd.

Kwaliteit kan hier worden onderzocht door een kleine steekproef te trekken uit niet gekoppelde en wel gekoppelde paren van records en deze vervolgens handmatig te onderzoeken met behulp van de overige variabelen in het record en daarbij te kijken naar bijvoorbeeld (afwijkingen op) kengetallen (vergelijk: uitbijters). Hierbij zijn (schattingen van) koppelfouten van de 1^e en de 2^e soort te gebruiken als kwaliteitsmaten.

5.6 Variant

Een variant op deze methode is die waarbij niet rechtstreeks wordt gekoppeld op de primaire sleutel, maar waarbij sprake is van een relatie- of koppelbestand. In dat geval kunnen verschillende eenheden toch worden gekoppeld.

6. Koppelen op secundaire sleutels, zonder koppelgewichten

6.1 Korte beschrijving

Bij het koppelen op basis van een secundaire sleutel worden één of meerdere variabelen gebruikt die het record mogelijk identificeren. Deze identificatie is, anders dan met behulp van een unieke primaire sleutel, niet per se ondubbelzinnig. Het probleem is dat een eenheid die uniek is op een set secundaire sleutels, niet per se uniek hoeft te zijn in de populatie. In de populatie van Nederlanders zijn er diverse personen die 'Janssen' als achternaam hebben, of die van beroep 'ambtenaar' zijn. Zelfs bij combinaties van enkele van zulke variabelen kan men nog dubbelen hebben: er zijn meerdere ambtenaren die Janssen heten. Daarentegen zou iemand die 'Wladimirow' heet en advocaat is wel uniek kunnen zijn in Nederland. Hoe meer van dergelijke *direct of indirect identificerende variabelen*[14] men ter beschikking heeft (zoals initialen, voornaam, familienaam, bedrijfsnaam, geslacht, geboortedatum, leeftijd (op een bepaald moment), adres, beroep, etc.) des te groter is de kans dat er unieke personen worden aangeduid in een bestand. Ze hoeven niet allemaal uniek te zijn, maar een deel van de personen gerepresenteerd in een bestand zou uniek kunnen zijn. Hoe meer er van dergelijke variabelen in een bestand aanwezig zijn, hoe meer unieken (in de populatie) en niet alleen in het bestand men aantreft. Indien de scores betrouwbaar zijn, zijn dit dan waarschijnlijk ook in werkelijkheid populatie-unieken.

Daarbij komt dat er ook nog waarnemingsfouten en andere afwijkingen kunnen voorkomen in de waarden die deze indirecte identificatoren aannemen. In dat opzicht verschillen ze van variabelen die als primaire sleutelvariabele gelden. Bovendien zijn 'waarden-met-afwijkingen/fouten' op secundaire sleutels meestal nog bruikbaar om te koppelen. Fouten op scores van primaire sleutels zijn meestal onbruikbaar (denk aan BSN-nummers met typefouten).

6.2 Toepasbaarheid

Voorwaarde om deze methode toe te kunnen passen is dat in beide koppelbestanden gemeenschappelijke indirect identificerende variabelen[15] aanwezig zijn, op basis waarvan de koppeling kan worden uitgevoerd. We laten daarbij ook toe dat twee overeenkomstige variabelen ook een verschillend domein hebben, bijvoorbeeld met een andere categorie-indeling (bijvoorbeeld leeftijd in 5-jaarsklassen in het ene bestand en 10-jaarsklassen in het andere). Verder laten we toe dat waarnemingsfouten kunnen voorkomen op de scores van deze variabelen.

---

[14] Deze namen zijn ontleend aan de statistische beveiliging. Zie Willenborg en De Waal (2000). Overigens zijn primaire sleutels als sofinummer of BEID hier niet bedoeld. Wel echter variabelen als familienaam, initialen, voornaam, adres, etc. Dit zijn weliswaar directe identificatoren, ze duiden geenszins altijd unieke eenheden aan. Immers een naam als Janssen komt veel voor, evenals woonplaats Amsterdam, of adres Dorpstraat. Gecombineerd zijn ze veel krachtiger en kunnen ze unieke eenheden als personen gaan aanduiden. Een score of BSN duidt al een unieke persoon aan.

[15] Dat ze gemeenschappelijke variabelen hebben is niet voldoende. Het zou dan bijvoorbeeld kunnen gaan over variabelen die een opinie of opvatting uitdrukken. Bijvoorbeeld antwoorden op vragen als: op welke partij heeft u de laatste keer gestemd? Voelt u zich veilig op straat na zonsondergang? De antwoorden op dit soort vragen zijn in het algemeen niet zo betrouwbaar.

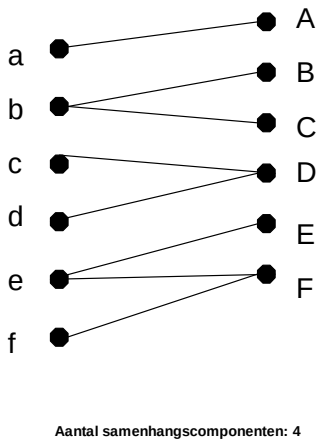
De keuze om met een koppelmethode te werken die geen koppelgewichten gebruikt zal in de praktijk vaak samenhangen met performance. Als de te koppelen bestanden groot zijn, werken deze methoden in het algemeen sneller dan die met koppelgewichten. Te verwachten is dat de kwaliteit van de gerealiseerde koppelingen - in termen van koppelfouten van de eerste en tweede soort - in het algemeen lager is.

6.3 Uitgebreide beschrijving

Een KK-graph geeft aan welke records op grond van het gebruikte koppelcriterium kandidaten zijn om te koppelen. Het opstellen van een KK-graph komt tot stand na het toepassen van het koppelcriterium op ieder mogelijk tweetal records.

Een KK-graph is formeel een bipartiete graph en is als volgt gedefinieerd voor een koppelprobleem, waar twee bestanden A en B bij betrokken zijn, en een koppelcriterium K wordt gebruikt op een koppelsleutel S. Voor een voorbeeld van een KK-graph, zie figuur 6.1.

Figuur 6.1: Voorbeeld KK-graph zonder koppelgewichten



De files A en B vatten we op als verzamelingen records. Zij $G = (V, E)$ de KK-graph bij dit koppelprobleem. De puntenset V is gegeven door $V = A \cup B$ en de kantenset E bestaat uit de paren $\{a, b\}$ met $a \in A, b \in B$ die bovendien aan het koppelcriterium K voldoen.

In figuur 3 is een KK-graph getekend. De kanten geven de koppelkandidaten aan. De koppeling $\{d, h\}$ is de enige die ondubbelzinnig te maken is, los van het gebruikte koppelcriterium. Afhankelijk van het koppelcriterium zijn er nog meer koppelingen te maken. Indien het een 1:1-koppeling betreft zijn er nog twee extra koppelingen mogelijk, namelijk:

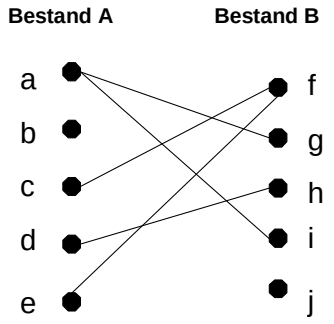
1. $\{a, g\}$ of $\{a, i\}$ (één van beide)

2. {c,f} of {e,f} (één van beide).

De keuzes bij 1. en 2. zijn onafhankelijk van elkaar te maken.

In het geval van KK-graph zonder gewichten tellen alle kandidaat-koppelingen even zwaar. Een belangrijk voorbeeld van een koppelcriterium dat leidt tot een KK-graph zonder koppelgewichten is dat van gelijkheid van scores op de koppelsleutel. De koppelwijze gebaseerd op dit criterium wordt ook wel aangeduid als *exact koppelen* of *exact matchen*. Opgemerkt zij dat dit louter slaat op het feit dat het gebruikte koppelcriterium exacte gelijkheid eist van scores op de variabelen in de koppelsleutel om de bijbehorende records als koppelkandidaten te beschouwen. Het heeft niets te maken met 'nauwkeurigheid', of het 'foutloos' zijn van de koppelingen. De reden hiervoor is dat in de praktijk fouten, afwijkingen of onregelmatigheden voorkomen in de te koppelen bestanden, en meer in het bijzonder op de koppelsleutel. Deze fouten in de gegevens leiden er toe dat koppelkandidaten worden gevonden die geen betrekking hebben op dezelfde eenheden. Maar ook dat koppelingen worden gemist.

Figuur 6.2: KK-graph zonder koppelgewichten



Wanneer we het koppelcriterium voor exact koppelen verruimen kunnen we wel eenheden bij elkaar zoeken met een gering aantal afwijkingen in de scores op de gebruikte koppelsleutel. Stel dat de koppelsleutel bestaat uit de variabelen (secondary keys) $s_1,...,s_k$. Stel dat a een record is uit bestand A en b een record uit bestand B. De scores van a en b geven we aan als respectievelijk $(s_1^a,...s_k^a)$ en $(s_1^b,...s_k^b)$. De records a en b zijn koppelkandidaten als $(s_1^a,...s_k^a)=(s_1^b,...,s_k^b)$. We zouden in plaats daarvan kunnen eisen dat er afwijkingen mogen zijn, maar een beperkt aantal, zeg maximaal p. Als we bijvoorbeeld de Hamming-afstand d_H (zie ook hoofdstuk 3) zouden

gebruiken, zouden we records a en b als koppelkandidaten kunnen beschouwen als $d_H(a,b) \leq p$, en als niet gekoppeld als $d_H(a,b) > p$. Iets dergelijks zou ook met een andere metriek gelden.

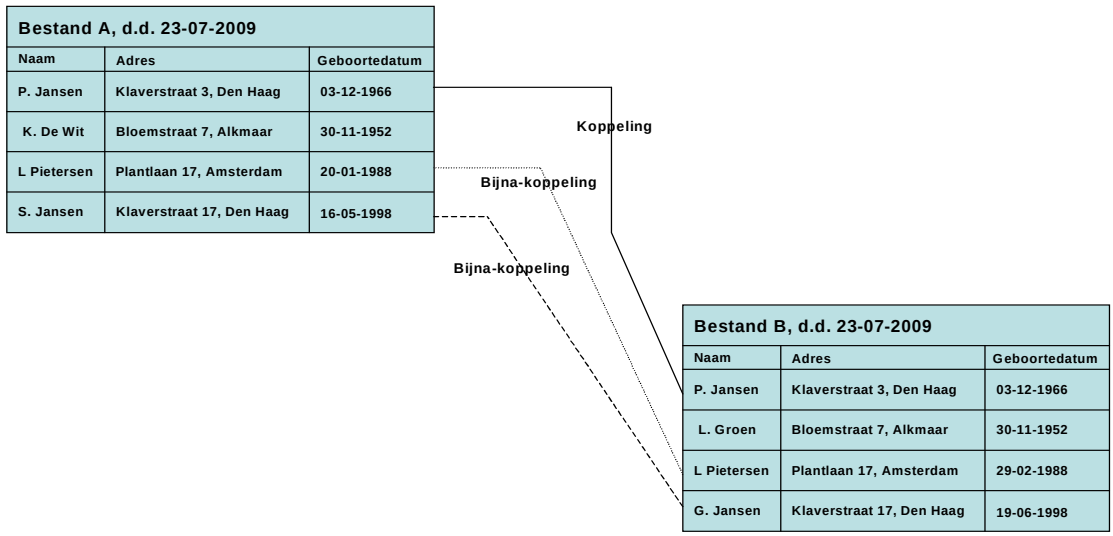
De hierboven beschreven aanpak is wat zwart-wit: twee records zijn koppelkandidaten of niet. We zouden nog een twijfelzone kunnen invoeren. In geval van de Hamming-afstand gebruiken we dan twee parameters (natuurlijke getallen) p,q met $p < q$. We volgen dan het volgende beslissingsschema:

Tabel 6.1 Koppelkandidaten, twijfelgevallen, geen koppelkandidaten

$d_H(a,b) \leq p$	betekent	a en b zijn koppelkandidaten
$p < d_H(a,b) \leq q$	betekent	a en b zijn twijfelgevallen, en dienen te worden geïnspecteerd door een inhoudsdeskundige die bepaalt of a en b wel/geen koppelkandidaten zijn
$d_H(a,b) > q$	betekent	a en b zijn geen koppelkandidaten

De parameters p en q kunnen zó gekozen worden dat veel of weinig koppelkandidaten geïnspecteerd moeten worden, afhankelijk van de beschikbare capaciteit van deskundigen die de twijfelgevallen kunnen beoordelen. De parameters p en q zijn voorbeelden van cut-off-waarden.

Figuur 6.3: Koppelen m.b.v. een Hamming-metriek



6.4 Voorbeeld

Men heeft twee bestanden A en B en wil deze op basis van de gemeenschappelijke koppelvariabelen naam, adres en geboortedatum (alle secundaire sleutels) koppelen. In eerste instantie koppelen alleen de eerste records van beide bestanden (P. Jansen). Er is sprake van een koppeling op basis van gelijkheid van de scores van alle koppelvariabelen. In tweede instantie wordt de eis van gelijke scores op de koppelvariabelen naam, adres en geboortedatum afgezwakt. Bij de variabele geboortedatum is het voldoende om alleen te koppelen op het geboortejaar. Dat levert een extra koppeling op bij de naam L. Pietersen. Ten slotte zwakt men de eis nog verder af. Er is al sprake van een koppeling als de achternaam, het geboortejaar en het adres gelijk zijn. Dat levert nog een extra koppeling op namelijk die tussen S. Jansen (bestand A) en G. Jansen (bestand B). In figuur 6.3 is deze situatie weergegeven. Dit is een voorbeeld waarbij we ook zouden kunnen stellen dat een metriek is gebruikt, namelijk een Hamming-metriek (zie paragraaf 7.3.1.1). Maar deze metriek leidt tot gewichten 0 (geen koppelkandidaat) of 1 (koppelkandidaat).

Vergelijkbare voorbeelden zijn er ook bij de economische statistieken. Bijvoorbeeld als gekoppeld wordt op basis van koppelvariabelen als ‘Naam van het bedrijf’, ‘adres’ en ‘telefoonnummer’. Het mag duidelijk zijn dat dit geen gemakkelijke opgave is omdat de namen van bedrijven op veel verschillende manieren kunnen worden vastgelegd. Zo kan het de ene keer gaan om de formele rechtspersoon (bijv. Verkoop Vanalles BV), de andere keer om een afkorting (bijv. Vanalles) en de volgende keer om de naam van de eigenaar (bijv. G. Jansen).

6.5 Kwaliteitsindicatoren

Ook hier zijn aantallen miskoppelingen of gemiste koppelingen te gebruiken als kwaliteitsmaten. Er spelen hier enkele zaken die corresponderen met de cruciale stappen in een koppelproces:

1. Het vinden van koppelkandidaten. Hierbij spelen een rol:
    a. Het gebruikte koppelcriterium (bijvoorbeeld gebruik makend van de Hamming-afstand) om records wel/niet als koppelkandidaten te beschouwen.
    b. In geval men een metriek etc. gebruikt, is het de vraag in hoeverre deze adequaat het onderliggende foutproces verdisconteert. (Zie ook paragraaf 7.3.) Ook de keuze van cut-off-waarden is van invloed op welke records als koppelkandidaten worden beschouwd.
    c. Eventueel gebruikte blocking variabelen (bijvoorbeeld bij grote bestanden); door de koppelbestanden te partitioneren en de zoekruimte bewust te beperken (vanwege de performance) kan het zijn dat men kandidaat-koppelingen mist, en uiteindelijk dus ook koppelingen.
2. het selecteren van de uiteindelijke koppelingen uit de koppelkandidaten. Ook hier wordt een criterium gebruikt. De vraag is in hoeverre dit tot correcte keuze leidt.

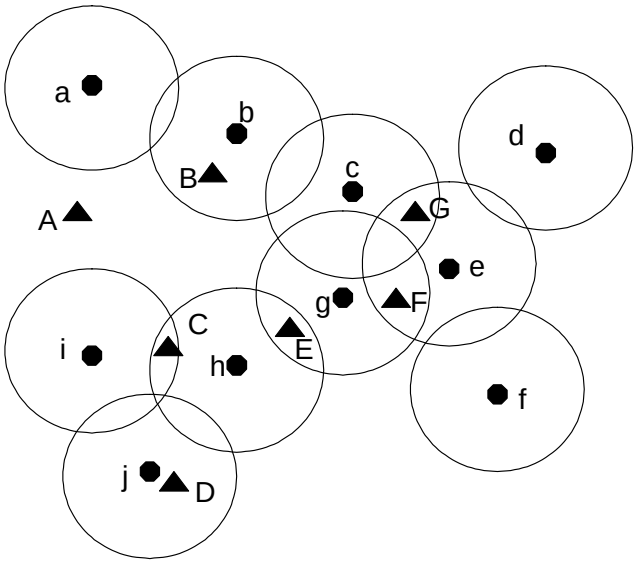
De kwaliteit van een gebruikte koppelingsmethode is te schatten op basis van inspectie van koppelingen van proefbestanden. Dat is hier arbeidsintensief. Niet alleen moet worden gekeken naar de koppelkandidaten en de uiteindelijk geselecteerde koppelingen, maar ook naar eventueel gemiste koppelingen bij diverse parameterinstellingen.

6.6 Variant: gebruik van een afstandsfunctie

Bij koppelen op basis van een primaire sleutel wordt geëist dat records exact dezelfde score hebben op de gebruikte koppelsleutel. Dit criterium kunnen we ook gebruiken bij het koppelen met behulp van secundaire sleutels. Deze methode is hier minder aantrekkelijk. We kunnen deze eis afzwakken en twee records als koppelbaar beschouwen, als de scores voor ten minste k (in te stellen parameter) van de maximaal n (lengte van de koppelsleutel = aantal koppelvariabelen in de koppelsleutel) gelijk zijn. In feite wordt hier gebruik gemaakt van een metriek, namelijk de zogenaamde Hamming-afstand of Hamming-metriek. Hier volstaan we met enkele opmerkingen die betrekking hebben op de situatie in dit hoofdstuk. Indien we de Hamming-metriek aan duiden met d_H [16] en we de scores op de koppelsleutel opvatten als vectoren van lengte n, dan komt het koppelcriterium dat hier gebruikt wordt feitelijk neer op:

Voor $\alpha \in A, \beta \in B$ zijn α en β koppelkandidaten dan en slechts dan als $d_H(\alpha, \beta) \leq k$.

Figuur 6.4: Records uit twee bestanden als punten gerepresenteerd en omgevingen van records uit één van de bestanden



Bolletje: record uit bestand A; driehoekje: record uit bestand B; cirkel: omgeving van record uit A. Kies bij voorkeur het bestand met 'harde' koppelgegevens als het bestand met records waar omgevingen. Van worden bepaald om te zien of er koppelkandidaten in voor komen. De keuze van de straal van de bollen is nog een interessant praktisch punt: hoe groter hoe meer kandidaten, maar ook hoe meer foute koppelingen voorkomen.

We kunnen het zo formuleren, dat alle β uit B die in een bol met straal k (gemeten met behulp van d_H) rond α zitten, alle koppelkandidaten voor α leveren. Zie ook figuur 6.4. Voor iedere α in A kunnen we dit nagaan in B. (Of omgekeerd, voor iedere β in B kunnen we nagaan welke α in A in een bol met straal k om β zitten. Dat levert hetzelfde resultaat op). Merk op dat we hier alleen gebruiken of een record in een bol rond een 'punt' zit, en niet wat de afstand precies is. Dat laatste

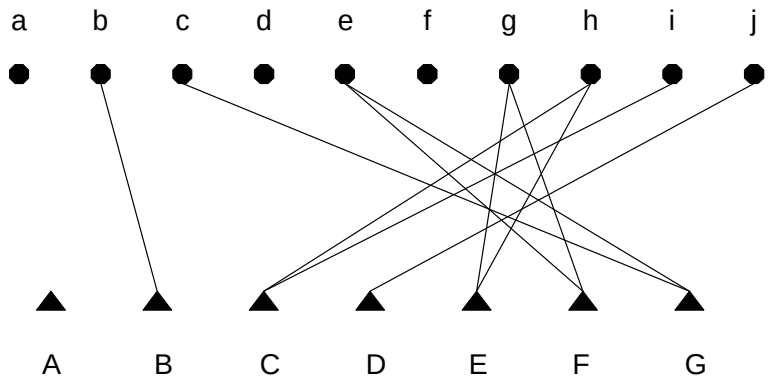
[16] In deze notatie is de lengte van de koppelsleutel, n, bewust weggelaten, om de notatie eenvoudig te houden.

zouden we wel kunnen gebruiken om records die dichter bij dit ‘middelpunt’ zitten een hoger koppelgewicht te geven. Dat gebeurt in hoofdstuk 7.

Indien men het voorgaande kritisch beziet, kan men concluderen dat de keuze van een Hamming-afstand niet wezenlijk is; men kan net zo goed een andere metriek kiezen om tot een koppelcriterium te komen. Dus op basis van een metriek d is het mogelijk een koppelcriterium te formuleren:

Voor $\alpha \in A, \beta \in B$ zijn α en β koppelkandidaten dan en slechts dan als $d(\alpha, \beta) \leq k$.

Figuur 6.5: KK-graph van situatie in het vorige plaatje



Stel 1:1 koppeling. Koppeling bB wordt zeker gemaakt. Bij cG, eF,eG, gF zijn twee keuzes mogelijk: {cG,eF} of {eG,gF} Zonder verder informatie is een voorkeur niet uit te motiveren

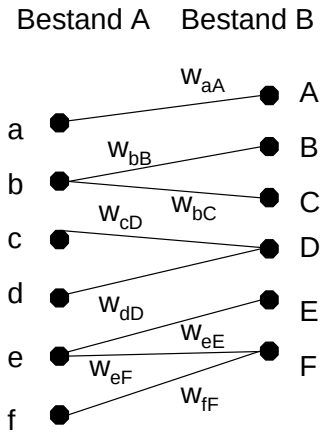
Ook nu kunnen we het zo formuleren, dat alle β uit B die in een bol met straal k rond α zitten (gemeten met d), alle koppelkandidaten voor α leveren. Voor iedere α in A kunnen we dit nagaan in B. (Of omgekeerd, voor iedere β in B kunnen we nagaan welke α in A in een bol met straal k om β zitten). Dit alles levert een KK-graph op, waar records uit A en B zijn gepresenteerd en die welke koppelbaar zijn, door een kant met elkaar verbonden. Zie figuur 6.5.

7. Koppelen op secundaire sleutels, met koppelgewichten

7.1 Korte beschrijving

Diverse koppeltechnieken maken gebruik van koppelgewichten, die gebruikt kunnen worden om te differentiëren tussen de verschillende potentiële koppelingen die in de KK-graph bij een koppelprobleem bestaan. Zie figuur 7.1. De reden om met koppelgewichten te werken kunnen divers zijn: men wil tot uitdrukking brengen dat niet alle variabelen even betrouwbaar zijn (dat wil zeggen betrouwbare scores hebben). Het kan zijn dat men wil aangeven dat de eenheden die corresponderen met records die koppelkandidaat zijn, in een bepaalde mate op elkaar lijken (een bepaalde mate van (dis)similarity vertonen). Of, men wil aangeven dat ze een bepaalde afstand tot elkaar hebben, gemeten volgens een bepaalde metriek. Of men gebruikt een kans om aan te geven dat twee eenheden dezelfde zijn. Daarbij wordt een kansmodel gebruikt om verschillen in scores op de koppelsleutel te kwantificeren.

Figuur 7.1: KK-graph met koppelgewichten



Hoe lager/hoger de koppelgewichten, hoe meer/minder de verbonden eenheden bij elkaar liggen of op elkaar lijken. Wat het verband tussen de grootte van koppelgewichten en de mate van gelijkheid is, moet per probleem worden vastgesteld.

7.2 Toepasbaarheid

De koppelmethode zonder koppelgewichten zou men zwart-wit kunnen noemen: twee records zijn koppelkandidaten of niet. Er is daar geen plaats voor nuance. Er zijn echter situaties waar het wenselijk is deze nuance wél aan te brengen. Denk hierbij aan variabelen als voornaam, achternaam, straatnaam, plaatsnaam etc. Hier wil men tot uitdrukking kunnen brengen in welke mate twee achternamen zeg, van elkaar verschillen. Het verschil tussen ‘Jansen’ en ‘Janssen’ is kleiner dan het verschil tussen ‘Jansen’ en ‘Cuypers’. Het gaat hier dus zuiver om de spelling van de namen, de letters die er in voorkomen en de volgorde waarin ze staan. Dit is echter te kwantificeren, met behulp van een metriek (zie paragraaf 7.3.1.1). In andere voorbeelden gaat het

niet zozeer om de mate waarin strings van elkaar verschillen, maar om de mate waarin de betekenissen van de strings verschillen van elkaar. Dit is bijvoorbeeld het geval bij beroepen. De woorden (begrippen) ‘leraar’ en ‘docent’ verschillen behoorlijk van elkaar als men naar de letters kijkt die erin voorkomen, maar qua betekenis liggen ze dicht bij elkaar, of kunnen zelfs als gelijk worden beschouwd. Het gaat om een ander afstandsbegrip dan het hiervoor besproken afstandsbegrip. Het gaat nu om de betekenis of semantiek geassocieerd met de strings opgevat als woorden of begrippen. Eenzelfde verschil krijgen we als we niet op de schrijfwijze van strings letten, maar op de uitspraak. Namen als ‘Taylor’ en ‘Teler’ liggen fonetisch gezien dicht bij elkaar. In beide gevallen meten we niet de afstand van twee strings s,t met behulp van een metriek d, dus $d(s,t)$, maar van $D(f(s), f(t))$, waar $f : S \to T$ een afbeelding is van de verzameling S van strings naar een ruimte T van betekenissen, of klanken etc. met D een metriek op T.

Een metriek is een voorbeeld van een functie die gebruikt kan worden om koppelgewichten uit te rekenen. Deze koppelgewichten kunnen gebruikt worden om de sterkte van een kandidaat-koppeling tot uitdrukking te brengen. In de praktijk hoort hier nog bij dat men met cut-off-waarden moet werken: koppelingen die te zwak zijn in termen van het bijbehorende koppelgewicht worden beschouwd als zijnde geen koppelkandidaten. Het is de kunst om dergelijke cut-off-waarden goed in te stellen: niet zodanig dat men te veel irrelevante matches mee neemt, maar wel zodanig dat de correcte matches niet gemist worden. In de praktijk vergt dit experimenteren met diverse instellingen van de cut-off-waarden.

Andere mogelijkheden om tot koppelgewichten te komen dan met behulp van metrieken worden in paragraaf 7.3.1 besproken. Alle overwegingen om koppelgewichten te gebruiken dienen te zijn ingegeven door de processen of mechanismen die tot verschillen in de data aanleiding hebben geven, of zouden kunnen geven. Dat kunnen verschrijvingen zijn (‘Jansen’ in plaats van ‘Janssen’, of het gebruik van alternatieve aanduidingen als die vrijheid bestaat (bij adressen: ‘Dorpsstr.’ in plaats van ‘Dorpsstraat’; bij beroepen: ‘docent’, ‘leerkracht’, ‘onderwijzer’, ‘leraar’ duiden allemaal vergelijkbare functies aan in het onderwijs). Men dient daarom een grondige kennis te hebben over de wijze waarop de te koppelen bestanden zijn samengesteld. Daarnaast kan het zijn dat niet precies dezelfde koppelvariabelen worden gebruikt in beide bestanden, of dat de scores niet betrekking hebben op hetzelfde moment in de tijd. Hierdoor kunnen kenmerken van een entiteit (individu, bedrijf, etc.) veranderd zijn.

7.3 Uitgebreide beschrijving

7.3.1 Berekening van koppelgewichten

Er zijn verschillende manieren om koppelgewichten te bepalen die gebruikt kunnen worden in een koppelprobleem. Een aantal van die manieren bespreken we hier. De opsomming is niet uitputtend, maar geeft wel een aantal belangrijke voorbeelden. Deze koppelgewichten worden gebruikt bij het koppelen, als de informatie over de ‘koppelkandidatuur’ van twee records niet tweewaardig wordt weergegeven (‘wel’, ‘niet’ koppelkandidaat) maar met meer nuance. De mate waarin twee records bij elkaar passen kan in een koppelgewicht tot uitdrukking worden gebracht.

In de bespreking in de onderstaande paragrafen beschouwen we steeds twee bestanden, A en B, met records, waarvoor er gemeenschappelijke koppelvariabelen $v_1,...,v_n$ bestaan die samen de koppelsleutel vormen, op basis waarvan de records in beide bestanden gekoppeld worden.

7.3.1.1 Op basis van metrieken of gegeneraliseerde metrieken

Eerder in dit rapport (hoofdstuk 3) hebben we metrieken en gegeneraliseerde metrieken in algemene zin besproken. Daar was vooral van belang welke eigenschappen een metriek, en diverse van zijn varianten hebben. Voor het koppelen is het van belang geschikte metrieken te vinden voor ieder van de variabelen in een (secundaire) koppelsleutel, of beter voor ieder van de typen variabelen. Als variabelen in een secundaire koppelsleutel treden op: namen (voornamen, familienamen, bedrijfsnamen, straatnamen, gemeentenamen, etc.), tijdsaanduidingen (geboortedatums, leeftijden op een bepaald referentietijdstip), geslachtsaanduiding, burgerlijke staat, beroepen, etc. Deze problematiek kan als een apart deelterrein binnen het koppelen worden beschouwd.

Een algemeen probleem is om bij gegeven structuren nieuwe structuren te vinden. Bij verzamelingen met metrieken geldt hetzelfde. Bij een gegeven verzameling X met metriek d – het paar (X, d) wordt ook wel een metrische ruimte genoemd – geldt voor iedere $Y \subseteq X$ dat $d|_{Y\times Y}$, dus de restrictie van d tot $Y \times Y$, een nieuwe metrische ruimte $(Y, d|_{Y\times Y})$ oplevert.

Een ander voorbeeld is om een productverzameling te voorzien van een metriek, als alle component-verzamelingen voorzien zijn van een metriek. Stel dus dat we een aantal variabelen hebben met op de bijbehorende domeinen een metriek gedefinieerd. Zij $X = D_1 \times ... \times D_n$, waar D_i het domein is van koppelvariabele v_i met metriek d_i, dan is een metriek d_w op X bijvoorbeeld te definiëren als

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

voor zekere constanten $a_i > 0$ voor $i = 1,..., n$. In de praktijk kunnen de gewichten a_i gebruikt worden om het relatieve belang van de metrieken-per-variabele op elkaar af te stemmen. De variant die in paragraaf 6.6 wordt beschreven, maakt ook gebruik van een metriek, alhoewel dit misschien op het eerste gezicht niet zo op valt.

Weer een andere manier om een metriek op X te definiëren is als volgt:

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

In dit geval tellen de deelmetrieken allemaal even zwaar mee. We kunnen hier ook differentiëren in de zwaarte van de deelmetrieken door met gewichten $a_i > 0$, $i = 1,..., n$ te definiëren:

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

Een metriek die voor iedere variabele te definiëren is, is de 'zwart-wit metriek' d_{01}, gedefinieerd als

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

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

Op basis van deze zwart–wit metriek per variabele is de Hamming-afstand te definiëren

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

Dat d_{01} en d_H 'universeel' te gebruiken zijn – dus ongeacht het type varabele(n) – is een kracht zowel als een zwakte. In de praktijk is er juist behoefte aan metrieken die zijn toegesneden op specifieke variabelen, of typen variabelen. Deze metrieken brengen ook meer nuance aan, in de zin dat ze sommige verschillen als meer of minder kwantificeren dan andere.

We geven hier en in Appendix B enkele voorbeelden van dergelijke metrieken. Het doel is vooral om een aantal mogelijkheden te illustreren.

Een belangrijk geval betreft variabelen waarin alfanumerieke strings de scores vormen, zoals bij naam (voornaam, familienaam, straatnaam, bedrijfsnaam, etc.). Als we zo'n naam opvatten als een string van symbolen zouden we twee strings σ en τ kunnen vergelijken via een Levenshtein-afstand(sfunctie) d_L. Voor meer informatie over deze metriek, bij toepassingen in een aantal vakgebieden zoals de biologie, de linguïstiek en de bio-informatica, zie bijvoorbeeld Sankoff en Kruskal (1983, pp. 18,19) Zo is bijvoorbeeld $d_L(Janssen, Jansen) = 1$ (verwijder 's' uit de eerste string) en $d_L(Hendricks, Hendrikx) = 2$ (verwijder 'c' uit de eerste string en verander vervolgens 's' in 'x').

Deze definitie van afstand is gebaseerd op de geschreven tekst (naam). Het is soms echter beter rekening te houden met de uitspraak van de tekst (naam). In de praktijk kunnen spellingsvarianten als "Janse', 'Jansse', 'Jansen', 'Janssen', 'Janszen' en 'Janzen' voorkomen en 'Hendriks', 'Hendricks', 'Hendriksz', 'Hendrix', 'Hendrikx', en 'Hendrickx'. Men zou hier eigenlijk gebruik willen maken van een functie die deze namen fonologisch afbeeldt. In dat geval zouden de spellingsvarianten van 'Jansen', respectievelijk 'Hendriks' op eenzelfde beeld worden afgebeeld. Zie voor een, wat meer technische, vervolgdiscussie, Appendix B. Daar worden ook nog enkele andere metrieken voor strings besproken.

Om strings met elkaar te vergelijken wordt ook wel gebruik gemaakt van trigrammen. Trigrammen kunnen worden gebruikt om strings met een (beperkt) aantal spellingsafwijkingen met elkaar te relateren. In het onderstaande voorbeeld illustreren we het begrip trigrammen.

Voorbeeld: Neem de namen '_Hendriksz_' en '_Heinrichs_' (aan het begin en einde van iedere string voegen we een spatie ('_') toe; we bekijken deze uitgebreide strings). De trigrammen voor de eerste string zijn (we noteren alles in lower case letters): (_he, hen, end, ndr, dri, rik, iks, ksz, sz_) en voor de tweede string (_he, hei, ein, inr, nri, ric, ich, chs, hs_) .

Merk op dat we hierboven de trigrammen in volgorde hebben laten staan, in rijen. We kunnen echter ook de bijbehorende verzamelingen bekijken, in dit geval {_he, hen, end, ndr, dri, rik, iks, ksz, sz_} voor de eerste string en {_he, hei, ein, inr, nri, ric, ich, chs, hs_} voor de tweede string. Op basis van deze verzamelingen trigrammen kunnen we eenvoudig een metriek afleiden, door het aantal trigrammen te tellen in beide strings dat uniek voorkomt. Als we twee van dergelijke verzamelingen trigrammen hebben, zeg S en T voor strings σ en τ, respectievelijk, dan is

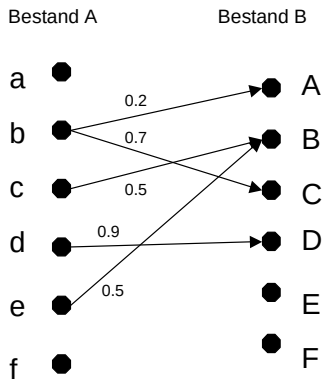
$$d_{tri}(\sigma,\tau) = |(S \cup T) \setminus (S \cap T)| = |S \setminus T| + |T \setminus S|$$

een metriek. We zouden nog een andere metriek kunnen definiëren die rekening houdt met het aantal overeenkomstige karakters in de set unieke trigrammen van beide strings. We gaan daar echter niet verder op in, omdat het te ver voert.

Voor variabelen waarbij het domein een natuurlijke (partiële) ordening heeft kunnen we de bijbehorende (gerichte) boom gebruiken om een afstand te bepalen tussen ieder tweetal punten in het domein. Deze is dan gelijk aan de lengte van het kortste pad in de boom dat beide punten verbind. Hierbij heeft iedere kant lengte 1.

Zo zijn er nog wel meer metrieken (en verwante functies) te bedenken voor koppelvariabelen. Dit kan men als een specialisme binnen het koppelen beschouwen. We gaan er hier echter niet verder

Figuur 7.2: KK-digraph met koppelgewichten, in dit geval kansen



We nemen aan dat de tweede betekenis van de vorige figuur geldt. Bij eenheid b in bestand A zijn er twee, namelijk A en C in bestand B, waar b uit geëvolueerd zou kunnen zijn. Verder is B in Bestand B een eenheid waartoe twee eenheden zich toe ontwikkeld zouden kunnen hebben in de tussentijd, namelijk c en e. De situatie zou dan betrekking kunnen hebben op twee bestanden met verschillende referentietijden. Merk op dat de kansen (vanuit b, of naar B) niet tot een optellen. Dat betekent dat er een positieve kans is dat b niet A of C hoeft te zijn. Bij B wordt de kans dat die uit c of e ontstaan is, 1 geacht. Een alternatief voor deze twee mogelijkheden wordt kennelijk uitgesloten.

op in, omdat het te ver af voert van de hoofdlijn van het betoog.

7.3.1.2 Op basis van kansen

Koppelgewichten kunnen ook gebaseerd zijn op kansmodellen. Stochastiek kan om verschillende redenen bij het koppelen zijn intrede doen. Zie figuur 7.2. We noemen de volgende:

1. Er kunnen fouten voorkomen in de secundaire koppelsleutels. Deze kunnen om uiteenlopende redenen daarin terecht zijn gekomen. Een antwoord op een vraag bij een enquête kan verkeerd begrepen zijn en dus foutief beantwoord door de desbetreffende respondent; een gegeven antwoord is foutief verwerkt, bijvoorbeeld verkeerd ingetoetst; bij het coderen van antwoorden zijn fouten gemaakt, etc. Dit soort fouten wordt collectief vaak aangeduid als niet-steekproeffouten. Men zou eerst alle belangrijke foutenbronnen moeten identificeren en modelleren met behulp van kansmodellen. Deze modellen kan men dan weer gebruiken om kansen te berekenen dat twee scores op overeenkomstige secundaire sleutels uit twee koppelbestanden bij elkaar horen.

2. De peilmomenten van de beide koppelbestanden verschillen dermate dat effecten van de dynamiek van de populatie merkbaar worden op de eenheden daarbinnen. Iemand kan in de tussentijd een jaar ouder zijn geworden; een bedrijf kan zijn gefuseerd, gesplitst of failliet zijn gegaan; iemand kan een ander beroep hebben gekregen; een werkloze kan een baan hebben gevonden, etc. Dus als de peilmomenten behoorlijk van elkaar verschillen is het niet vanzelfsprekend dat eenheden onveranderd zijn gebleven en/of hun scores op secundaire sleutelvariabelen.

3. Sommige, vergelijkbare koppelvariabelen in beide bestanden zijn niet helemaal hetzelfde gedefinieerd. De bijbehorende vraag kan anders zijn, of het waardenbereik van vergelijkbare variabelen kan iets verschillen. In dat geval kan het in sommige gevallen onduidelijk zijn welke scores bij elkaar horen. Stel dat {20,21} een leeftijdsklasse is in het ene koppelbestand en 11 - 20 en 21 - 30 in het andere bestand. De 20 en 21-jarigen komen in het eerste koppelbestand in dezelfde leeftijdsgroep terecht, maar in het tweede in twee verschillende leeftijdscategorieën. We kunnen ook nog wel schatten welk deel van de personen in de categorie (20,21) in het eerste bestand bij de leeftijdscategorie 11-20 en welk deel bij de leeftijdscategorie 21-30 terecht komt: $\frac{n_{20}}{n_{20}+n_{21}}$, respectievelijk $\frac{n_{21}}{n_{20}+n_{21}}$, waar n_{20} het aantal 20-jarigen is op de peildatum en n_{21} het aantal 21-jarigen op dat moment.

In de praktijk spelen vaak combinaties van deze oorzaken van verschillen. Bestanden kunnen andere peilmomenten hebben, er kunnen verwerkingsfouten in de data aanwezig zijn, en de eenheden hoeven niet precies vergelijkbaar te zijn. In paragraaf 7.4 wordt een voorbeeld gegeven van een situatie als in 3 hierboven en een voorbeeld dat een combinatie betreft van de punten 2 en 3 hierboven (variabelen met afwijkende waardenbereiken en verschillende peilmomenten).

7.3.1.3 Gewichten voor de kwaliteit van koppelvariabelen

In de praktijk zal men op basis van de kwaliteit van de scores willen kunnen differentiëren tussen de verschillende koppelvariabelen van de koppelsleutel. Sommige variabelen hebben nu eenmaal betrouwbaarder scores dan andere, en dit effect wil men mee laten wegen bij het bepalen van het overall koppelgewicht.

We beschouwen dit ‘kwaliteitsgewicht’ als een subjectief gewicht, dat een koppelaar instelt op grond van zijn kennis en ervaring met de diverse varabelen in de koppelsleutel. Het kan ook zo zijn dat eerst geëxperimenteerd moet worden om tot een goede keuze van deze gewichten te komen. Deze gewichten hebben alleen betekenis in hun onderlinge verhouding, niet in absolute zin.

Bij de discussie van multivariate metrieken in paragraaf 7.3.1.1 zijn gewichten ingevoerd die door de gebruiker kunnen worden ingesteld.[17] Een gebruiker kan het relatieve belang van een variabele ermee tot uitdrukking brengen voor de multivariate afstandsfunctie. Zo kan hij het effect van een

[17] Voor de wiskunde maakt het verder niet uit welke combinatie van gewichten wordt gekozen. Zolang alle gewichten >0 zijn is het resultaat een multivariate metriek.

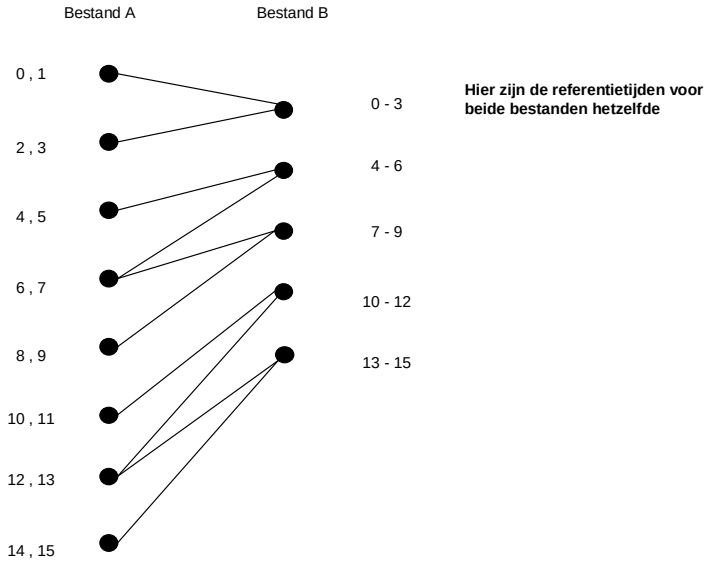
bepaalde variabele in het totaal beïnvloeden. Is de variabele betrouwbaar gemeten dan is een relatief hoog gewicht nodig. Is het een variabele met relatief meer fouten dan de andere variabelen in de koppelsleutel dan is het zaak deze variabele een lager gewicht te geven.

Overigens is het ook mogelijk het verschil in kwaliteit van koppelvariabelen anders tot uitdrukking te brengen, bijvoorbeeld door bij het koppelen scores af te werken in de volgorde van de kwaliteit van de koppelvariabelen (van hoog naar laag), en daarbij bepaalde afwijkingen in de scores met toenemende tolerantie te accepteren.

7.3.2 KK-graph met koppelgewichten

Indien we een methode hebben gekozen om koppelgewichten mee te bepalen, kunnen we aan de slag om een KK-graph met koppelgewichten te berekenen. Eventueel moeten we nog een cut-off-waarde hanteren zodat we kandidaat-koppelingen van twee records met een te laag koppelgewicht niet mee hoeven te nemen (dit worden geen kanten in de KK-graph).

Figuur 7.3. Twee leeftijdsvariabelen, één met 2-jaarsklassen (in koppelbestand A) en de ander met 3-jaarsklassen (in koppelbestand B)



7.4 Voorbeeld

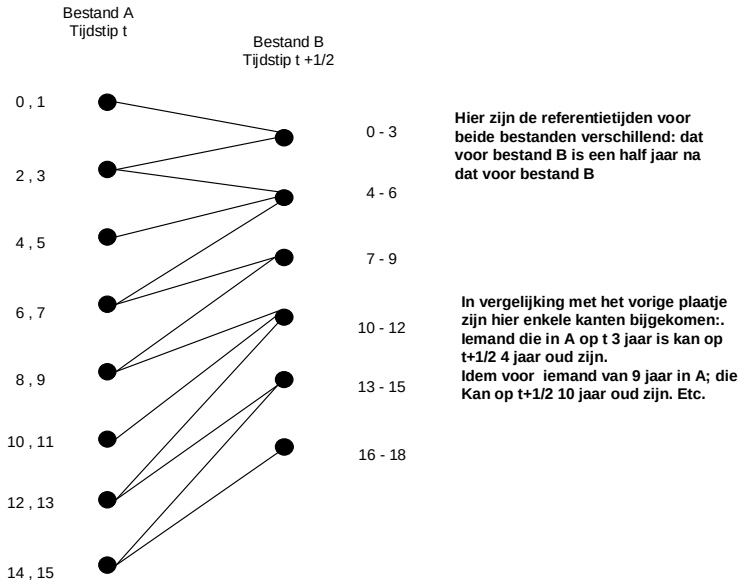
We bespreken hier een voorbeeld van een situatie waarbij twee koppelvariabelen overeenkomstig zijn, maar niet precies hetzelfde. In concreto gaan we uit van twee leeftijdsvariabelen.[18] De ene, die leeftijd in tweejaarsklassen specificeert komt voor in koppelbestand A en de andere, die leeftijden

[18] We veronderstellen dat in beide gevallen de leeftijd voorstelt: de leeftijd op het ijkmoment van het desbetreffende onderzoek. Bijvoorbeeld een antwoord op de vraag: ‘Hoe oud bent u nu?’

in driejaarsklassen weergeeft zit in koppelbestand B. Afhankelijk van de ijkmomenten voor beide bestanden (de tijd waarop de gegevens betrekking hebben) kunnen we de leeftijdscategorieën met elkaar in verband brengen. In figuur 7.3 is een graph getekend die de leeftijdscategorieën uit beide bestanden relateert voor het geval de ijkmomenten hetzelfde zijn.

In de praktijk hoeven de ijkmomenten van twee koppelbestanden helemaal niet precies gelijk te zijn. Sterker nog, dat zal eerder niet dan wel het geval zijn. Bovendien is er in de praktijk eerder sprake van een interval waarop de gegevens betrekking hebben dan van een moment (bijvoorbeeld omdat de individuen waar het om gaat niet allemaal op hetzelfde moment zijn geïnterviewd). In figuur 7.4 zijn de leeftijdscategorieën met elkaar in verband gebracht in het geval de ijkmomenten voor beide bestanden een half jaar verschillen. In dit geval kunnen sommige personen in het tussenliggende tijdsinterval een jaar ouder geworden zijn.

Figuur 7.4. Twee leeftijdsvariabelen, één met 2-jaarsklassen (in koppelbestand A) en de ander met 3-jaarsklassen (in koppelbestand B)



7.5 Kwaliteitsindicatoren

De kwaliteitsindicatoren die onder meer in paragraaf 6.5 zijn genoemd gelden hier ook, dus ten aanzien van de miskoppelingen en gemiste koppelingen. Van invloed daarop zijn de wijze waarop de gewichten worden berekend, het gebruik van cut-off-waarden en het gebruik van blocking variabelen om grote bestanden te stratificeren. Zie de discussie in paragraaf 6.5.

7.6 Varianten

In de paragrafen van dit hoofdstuk is de nadruk gelegd op de basisvariant voor koppelen met koppelgewichten, waarbij de koppelingen 1:1 zijn. Zoals eerder al is aangegeven zijn er echter ook

situaties mogelijk waar 1:n, m:1 en zelfs n:m koppelingen mogelijk zijn. Dit is het geval bij samengestelde eenheden als bedrijven die in de loop van de tijd kunnen splitsen of fuseren tot andere eenheden. Formeel betekent het dat de condities waaronder koppelingen mogelijk zijn, moeten worden aangepast. Ook gaat het niet om dezelfde eenheden, maar combinaties van eenheden die vergelijkbare entiteiten opleveren. Zie ook paragraaf 9.3.

In de bespreking tot nu toe is ook steeds verondersteld dat alle scores op secundaire sleutels aanwezig zijn. In de praktijk hoeft dit echter niet en kunnen er ook scores ontbreken. Het berekenen van koppelgewichten is in deze situatie dan lastiger, omdat de missing values niet zo maar weggelaten kunnen worden, maar vervangen moeten worden door stochasten, met een bekend veronderstelde verdeling. Met behulp van bijvoorbeeld het EM-algoritme moeten dan de onbekende parameterwaarden worden geschat. Voor informatie over het EM-algoritme zie Wikipedia (http://en.wikipedia.org/wiki/EM_algorithm) of de referenties die daar genoemd worden.

8. Koppelsoftware en IT-overwegingen

8.1 Koppelsoftware

Het koppelen van gegevens is in feite onmogelijk zonder gebruik te maken van één of ander softwarepakket. Niet alleen de grootte van de bestanden en de sets van koppelkandidaten, maar ook het opschonen en parsen van data in de pre-verwerkingsfase en het berekenen van gewichten en het bepalen van de subsets van gekoppelde en niet-gekoppelde records, vragen om een geautomatiseerde aanpak. Er zijn verschillende softwarepakketten op de markt waarmee gekoppeld kan worden. Bij de meeste gaat het om commerciële pakketen, die in toenemende mate als component zijn ingebed in pakketten voor het analyseren, opschonen en het matchen en presenteren van bedrijfsinformatie en data mining (bijv. Trillium). In dat geval zijn de gehanteerde methoden voor de gebruiker veelal een "black box". Een beperkt aantal pakketten is gebaseerd op open source (o.a. Febrl en the Link King). Bijna alle pakketten zijn gericht op het koppelen van persoonsgegevens. Er zijn geen pakketten die zich specifiek richten op het koppelen van bedrijfsgegevens. Elk van de koppelpakketten heeft zijn voor- en nadelen. Hier worden enkele mogelijkheden genoemd.[19]

- In de eerste plaats kan natuurlijk gekoppeld worden met *standaard software pakketten*, zoals MS Access, MS Excel, SQL, SPSS, Clementine, SAS[20] en Manipula. Deze pakketten zijn niet specifiek gemaakt om te koppelen en zijn dan ook alleen geschikt voor de eenvoudige vormen van koppelen, zoals joinen. Wil men ingewikkelder methoden gebruiken met bijvoorbeeld koppelgewichten dan zijn deze pakketten niet geschikt. Ook ondersteunen deze pakketten niet de activiteiten in de pre-verwerkingsfase, zoals parsen, blocking en specifieke vergelijkingscomponenten, zoals Soundex. Oracle en Microsoft hebben de intentie meer geavanceerde koppelsoftware als onderdeel van hun database management systemen op de markt te brengen;
- ook kan men eigen *maatwerk* maken. Dit vraagt echter om onderhoud en is vaak specifiek gericht op een bepaalde aanpak en methode. De flexibiliteit is beperkt;
- *Febrl (Freely Extensible Biomedical Record Linkage; www.sourceforge.net)*. Dit pakket is beschikbaar als freeware en open source (Pyton). Het pakket, dat specifiek is ontwikkeld voor het koppelen van data, kan dus desgewenst worden aangepast aan de eigen specifieke wensen en methoden. *Pakket:* het bevat voor de pre-verwerkingsfase onder meer mogelijkheden voor het editen, opschonen en parsen van data. Ook kunnen blocking-variabelen worden gebruikt en zijn er mogelijkheden voor ontdubbelen. Deze fase gaat gepaard met mogelijkheden om diverse overzichten van de data te maken. Er kunnen ook proefsets voor testruns worden gegenereerd. Het pakket beschikt over een GUI en is relatief gemakkelijk te gebruiken. Febrl omvat dus het gehele proces van koppelen en niet alleen de koppelfase zelf. *Methode:* met Febrl kan van verschillende koppelmethoden gebruik worden gemaakt (aan te passen door de gebruiker), waaronder met name die met gewichten (kansen op basis van Fellegi en Sunter; cut-off-waarden; op basis van o.a. Hamming, Soundex en Q-grams), met als resultaten matches, non-match en twijfelgevallen. Er zijn verschillende mogelijkheden om de gewichten te berekenen;

---

[19] Pakketten zijn niet uitgebreid bekeken. Er is in de meeste gevallen alleen gekeken op de site van de betreffende leverancier. Het gaat hier dus om globale beschrijvingen. Voor een volgende versie van deze nota is meer onderzoek gewenst, waarbij pakketten ten opzichte van elkaar kunnen worden afgewogen.

[20] SAS heeft wat uitgebreidere mogelijkheden om te koppelen. Zie het pakket Dataflux.

- Commerciële koppelsoftware, die bij het CBS beschikbaar is, betreft *Trillium (van Harte-Hanks; Trilliumsoftware.com)*. Trillium omvat het gehele koppelproces. Het moet gezien worden als een set van functies rondom een database management systeem. *Pakket:* Trillium beschikt over veel mogelijkheden voor de pre-verwerkingsfase (voor het verbeteren van de datakwaliteit; TS-Quality module) zoals een parser om de data op te schonen, ontdubbelen, te standaardiseren en een geo-codering systeem om adresinformatie te controleren (als ETL-tool). Daarnaast heeft het een module die kijkt naar inconsistenties in en over files heen (TS-Discovery). De gewenste bewerkingen kunnen via een GUI relatief gemakkelijk worden samengesteld (vergelijk Clementine). Trillium kan meer dan twee files tegelijkertijd koppelen. Er kan gebruik gemaakt worden van een samengestelde sleutel. Het resultaat wordt weggeschreven in drie classificaties, te weten: “pass”, “fail” of “query” (de twijfelgevallen). Het kan zowel on-line als in de batch gebruikt worden. Ook is er sprake van het bijhouden van alle acties en veranderingen in een audit trail. *Methode*: het product kent diverse mogelijkheden om te koppelen gebaseerd op koppelen met gewichten (Trillium parallel matcher; geen Fellegi en Sunter). Trillium bestaat al enige tijd (vanaf 1989) en is één van de commerciële leiders in dit gebied (zie Gartner, 2007[21]);
- *GRLS 3 (Generalized Record Linkage System;* http://www.statcan.gc.ca *)* is een commercieel koppelpakket van Statistics Canada en het is geschreven in C en werkt met name met Oracle als database. Het is specifiek opgezet voor gevallen dat er geen unieke primaire sleutel aanwezig is. Het is geschikt voor bestanden met zowel personen als bedrijven. *Pakket:* GRLS omvat twee stappen 1) bepalen van de koppelkandidaten op basis van door de gebruiker te bepalen criteria (beslisregels) en 2) bepalen of er sprake is van een koppeling met als resultaten: ”definite”, “possible” en “excluded”. Het is ook mogelijk om te ontdubbelen. Bij vergelijkingen tussen sleutelvariabelen kan bijvoorbeeld gebruik worden gemaakt van Soundex. Er kunnen proefbestanden worden gemaakt voor testruns. Het biedt de mogelijkheid één (voor ontdubbelen) of twee bestanden te koppelen. In de post-verwerkingsfase kunnen records ook worden samengeteld (“grouped”) en kunnen “possible” en “excluded” koppelkandidaten handmatig worden bekeken en verwerkt. *Methode*: het maakt gebruik van een koppelmethode met in te stellen gewichten met specifieke kansen (op basis van Fellegi en Sunter);
- *The Link King (Record linkage and consolidation software; www.the-link-king.com).* Evenals bij Febrl, betreft het hier freeware. Het is geschreven in SAS. *Pakket:* het systeem biedt de mogelijkheid voor ontdubbelen en blocking. Bij de vergelijkingen van de sleutelvariabelen zijn er o.a. mogelijkheden voor Jaro-Winkler string vergelijkingen, Soundex e.a., metrieken (bij bijvoorbeeld postcodes en strings), conversies van namen, gewichten van namen (Jansen heeft minder gewicht dan Walofski). Het heeft een GUI, waarin een specifieke component zit om te kijken naar de twijfelgevallen. Het pakket kan random een set genereren voor een proefrun. *Methodologie:* het pakket ondersteunt zowel methoden zonder gewichten en joinen als methoden met gewichten (kansen). De GUI biedt ondersteuning bij de keuze van de juiste methoden (op basis van Artificial Intelligence). Belangrijke nadelen zijn dat The Link King vraagt om een SAS-licentie en alleen persoonsrecords verwerkt;

[21] In hun eigen woorden: “Gartner, Inc. is the world's leading information technology research and advisory company. “. Zie: http://www.gartner.com/technology/home.jsp

- *SSA NAME3 (Search Software America; www.searchsoftware.com).* Met dit pakket is het mogelijk om bestanden met zowel personen, bedrijven als andere identifiers te koppelen. Het is een commercieel pakket ingebed in een systeem met componenten voor het verbeteren en presenteren van bedrijfsinformatie. *Pakket:* Ook dit pakket heeft routines om data in de pre-verwerkingsfase schoon te maken, te parsen, waaronder het standaardiseren van de sleutels en het aanmaken van subsets (blocking). De routines moeten wel ingebouwd worden in bestaande software. Het biedt de mogelijkheid om een set van records, bijvoorbeeld personen of bedrijfseenheden, te aggregeren naar andere eenheden, bijvoorbeeld huishouden en Onderneming. Het werkt zowel on-line als in de batch. *Methode:* het koppelen is gebaseerd op een methode met gewichten;
- *IQ-Matcher (Intech Solutions; www.intechsolutions.com.au). Pakket:* dit pakket biedt - net als andere commerciële pakketten vooral gericht op het geheel van bedrijfsinformatie - mogelijkheden voor het opschonen, standaardiseren en koppelen van data. Het ondersteunt ontdubbeling. Kan grote en meerdere bestanden verwerken. *Methode:* de koppelmethode betreft een methode met gewichten (kansen);
- *Link Plus (www.cdc.gov/cancer/npcr).* Is een stand alone koppelprogramma en het is freeware (de code is echter niet beschikbaar). *Pakket*: het pakket koppelt twee bestanden aan elkaar. Wat betreft de pre-verwerkingsfase biedt het alleen de mogelijkheid voor ontdubbelen. Het ondersteunt ook de handmatige verwerking van twijfelgevallen. Blocking is mogelijk. Er zijn vergelijkingsmogelijkheden op basis van naam. *Methode:* wat betreft de methode maakt het pakket gebruik van methoden met gewichten (kansen). Nadeel is dat het, zoals zoveel koppelsoftware, vooral gericht is op bestanden met personen;
- *Automach*. Het gaat hier om een commercieel pakket (onderdeel van Integrity; www.vality.com). *Pakket:* er zijn diverse mogelijkheden om data te controleren en te standaardiseren (pre-verwerkingsfase). Blocking is mogelijk. Er zijn verschillende mogelijkheden om strings met elkaar te vergelijken. *Methode:* het koppelen op basis van gewichten (kansen; Fellegi en Sunter);
- Andere opties zijn onder meer: GDriver (US Census Bureau/Winkler), Relais (Istat), LinkageWiz, Tailor (a record linkage toolbox), NameSearch van Intelligent Search Technology's, PA Oyster Engine, Fril, OxLink en Alta.

Omdat statistisch of synthetisch koppelen buiten de scope van dit rapport valt is hier niet gekeken naar software die deze methode ondersteunt.

8.2 IT-overwegingen

Het toevoegen van het koppelen van bestanden in (statistische) processen vraagt om het opzetten van een informatie-architectuur die past in de bestaande architectuur. In het geval van het CBS kan men daarbij denken aan: het opslaan van de resultaten in het DSC zodat ook anderen van de (tussen)resultaten gebruik kunnen maken en het benutten van standaard software eerder dan eigen maatwerk.

Ook al is software vereist voor het koppelen van data, dat ontslaat de gebruiker niet van de plicht om goed te begrijpen en te weten wat hij/zij aan het doen is. Koppelen kan een complex proces zijn. Daarbij dient echter te worden voorkomen dat het koppelproces voor de gebruiker een "black box" is. Dat risico bestaat vooral bij het gebruik van commerciële pakketten.

Het is van belang om de juiste vragen te stellen als men koppelsoftware wil selecteren. In Appendix C zijn hiervoor een serie overwegingen opgenomen.

9. Speciale onderwerpen

In dit hoofdstuk gaan we in op enkele speciale onderwerpen die bij het toepassen van koppelen in praktijksituaties spelen, of die met behulp van de in dit stuk beschreven koppelmethoden kunnen worden opgelost.

9.1 Grote bestanden

Dit levert problemen op bij het vaststellen van de koppelkandidaten, omdat er heel veel paren records moeten worden bekeken. Als we twee bestanden hebben, zeg A en B, dan is het aantal records dat zou moeten worden vergeleken $|A||B|$, dus het aantal records in A maal het aantal records in B. Als A en B beide een aantal records hebben van de orde 10^5 dan is de zoekruimte van de orde 10^{10} (= tien miljard) paren records. Een voor de hand liggende methode om hier mee om te gaan is door de koppelruimte (het aantal te inspecteren potentiële koppelingen) drastisch te verlagen. Dit kan bijvoorbeeld gebeuren door de koppelbestanden te partitioneren, bijvoorbeeld met behulp van een (koppel)variabele, de zogenaamde *blocking-variabelen*. In dat geval zijn technieken uit de data-mining interessant. Dit laatste betreft echter een betrekkelijk recente uitbreiding van de koppelingstheorie, waarvan de waarde nog moet worden aangetoond door nader onderzoek.

9.2 Bepaling van koppelparameters

Bij het toepassen van het koppelen op secundaire sleutels spelen enkele problemen die al eerder naar voren zijn gekomen maar die we hier nog een keer willen benadrukken. Bij het werken met koppelgewichten is het ook zaak om tot geschikte cut-off-waarden te komen. Anders zijn alle records uit het ene koppelbestand koppelbaar met alle andere records uit het andere bestand. Het is zaak op een geschikte manier aan te geven wanneer de koppelingen te zwak zijn om verder serieus te onderzoeken. Het andere probleem betreft het bepalen van één (multivariate) metriek in het geval een koppelsleutel uit meerdere secundaire sleutels bestaat, die ieder hun eigen metriek hebben. Het probleem is hier om met behulp van gewichten de metrieken-per-secundaire sleutel variabele te combineren tot één metriek. Deze problematiek is deels besproken in paragraaf 7.3.1.1.

9.2.1 Cut-off-waarden voor koppelgewichten

In de hoofdstukken 6 en 7 is sprake van cut-off-waarden die kunnen worden gebruikt bij het bepalen van kandidaat-koppelingen indien met koppelgewichten wordt gewerkt. Omdat het a priori lastig kan zijn een goede cut-off-waarde in te stellen, moet men ervan uitgaan dat eerst wat geëxperimenteerd moet worden met enkele proefkoppelingen om te komen tot een geschikte cut-off-waarde. Als de cut-off-waarde bekend is, kan men die paren records in de KK-graph meenemen waarvan de gewichten boven (zeg) deze cut-off-waarde zitten. (Men laat dan de minder sterkere potentiële koppelingen weg.) Indien men de cut-off-waarde te groot kiest en men wil experimenteren met een kleinere cut-off-waarde dan moet de KK-graph opnieuw worden berekend. Dat is niet handig. In dat geval kan men beter de KK-graph berekenen voor de kleinste cut-off-waarde die men wil bekijken. Voor grotere cut-off-waarden kan men dan gebruik maken van de berekende KK-graph met deze (kleinste) cut-off-waarde.

9.2.2 Gewichten in metrieken bij samengestelde koppelsleutels

In hoofdstuk 7 is aangegeven hoe men een multivariate metriek kan afleiden door een gewogen som te nemen van de metrieken per variabele. Voor alle waarden $a_i > 0$ voor $i = 1,...,n$ krijgt men zo formeel een metriek. De vraag is alleen hoe men de a_i's voor een concreet geval goed kiest. We kunnen deze vraag niet zonder meer beantwoorden. Daarvoor is meer onderzoek nodig. In de praktijk betekent het waarschijnlijk ook dat men met data zal moeten experimenteren en empirisch moeten bepalen welke combinatie goede resultaten geeft. Daarbij dient ook gelet te worden op de kwaliteit van de scores per variabele. Die hoeft niet hetzelfde te zijn voor de variabelen in één bestand, en in het bijzonder ook voor de secundaire keys.

9.3 Koppelen van gerelateerde eenheden

Tot nu toe hebben we in de koppelmethoden die zijn besproken verondersteld dat het de bedoeling is om informatie van dezelfde eenheden te koppelen. In de praktijk kan dat gecompliceerder liggen. Bij samengestelde eenheden bijvoorbeeld, zoals bedrijven, is het mogelijk dat deze transformeren: ze kunnen in de loop van de tijd splitsen of juist samengaan met andere, soortgelijke eenheden. Het kan ook zijn dat de informatie uit twee koppelbestanden betrekking heeft op twee verschillende referentietijden die voldoende uit elkaar liggen om mutaties in de eenheden zichtbaar te maken. Deze mutaties zijn een gevolg van de dynamiek die in veel[22] populaties aanwezig is. Als een eenheid in het ene koppelbestand nog als zodanig aanwezig is, maar in het andere alleen de gesplitste delen staan (of enkele daarvan) dan heeft het bij elkaar zoeken op gelijke eenheden geen zin: die zijn er simpelweg niet. Naarmate de referentietijden van beide bestanden verder van elkaar verwijderd zijn, is dit effect sterker. Verder spelen andere effecten ten gevolge van de dynamiek van de doelpopulatie een prominentere rol: nieuwe eenheden stromen in ('geboorte'), of juist uit ('sterfte'). Deze effecten spelen ook bij populaties bestaande uit 'atomaire', dat wil zeggen niet-samengestelde of enkelvoudige, eenheden, zoals personen. Indien er wordt gekoppeld op primaire sleutels levert deze dynamiek geen koppelproblemen op, mits bekend is hoe samengestelde eenheden uit elkaar geëvolueerd zijn. Maar indien er op secundaire sleutels wordt gekoppeld is er sprake van een complicatie. In ieder geval moet men erop bedacht zijn dat niet per se dezelfde eenheden aan elkaar worden gepaard, maar soms ook gerelateerde eenheden (eenheden die uit een andere eenheid ontstaan zijn). Koppeling is wel mogelijk als er sprake is van een koppeltabel, waarin de relaties tussen de eenheden in beide bestanden en over de tijd zijn vastgelegd, eventueel met de events die hebben geleid tot de mutaties.

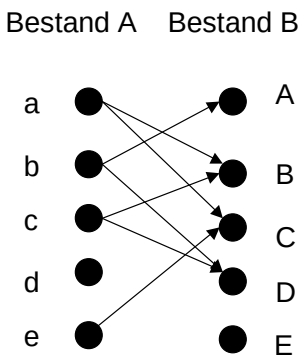
Ter illustratie van een koppelsituatie zoals in deze paragraaf bedoeld, is geen bipartiete digraph geschikt, maar een bipartiete gerichte graph, ofwel een bipartiete digraph. Een voorbeeld hiervan is te vinden in figuur 9.1.

---

[22] Bij populaties zoals die in dit stuk worden besproken allemaal. Het gaat hier om populaties van 'levende' objecten, die voortdurend veranderen. Er zijn wel statische populaties, zoals bijvoorbeeld: de 'populatie' van werken van een componist uit de barok. Maar ook hier bestaat de kans dat nieuwe werken worden ontdekt van die componist, of dat een werk dat eerder aan deze componist was toegeschreven later door een ander blijkt te zijn gecomponeerd. Maar dit zijn afbakeningsproblemen in plaats van mutaties zoals die in dit stuk bedoeld worden.

KK-digraphen kunnen gebruikt worden als koppel-graphen, maar in situaties waarbij er sprake is van een asymmetrie, bijvoorbeeld als de te koppelen bestanden op twee duidelijk verschillende momenten of perioden betrekking hebben. De richting van de pijlen kan dan de ontwikkeling aangeven. In plaats van een relatie als ‘is dezelfde eenheid als’ zoals die standaard is bij koppelen, kunnen we ook spreken van een relatie als ‘is ontstaan uit eenheid’. Net als de kanten bij KK-graphen kunnen de pijlen bij KK-digraphen wel/niet voorzien zijn van koppelgewichten.

Figuur 9.1: Voorbeeld van twee bestanden met verschillende referentietijden en hun onderlinge relatie



Interpretatie: a splitst in B en C. Of: eenheid a in bestand A kan geëvolueerd zijn in eenheid B of eenheid C in bestand B. NB Pijlen altijd van punten in bestand A naar punten in bestand B.

9.4 Wegwerken van restanten

In de praktijk kan het een probleem zijn om de records die niet gekoppeld worden zo maar weg te laten, omdat daardoor allerlei cijfers inconsistent kunnen worden. In dergelijke gevallen kan men besluiten om (alle of een deel van) de niet koppelbare records alsnog met elkaar te koppelen. Dat kan zelfs in het geval het risico groot is dat de koppelingen die dan verkregen worden niet dezelfde eenheden beschrijven. Wat dat betreft zit men dan in een vergelijkbare situatie van statistisch koppelen, in die zin dat eenheden gekoppeld worden die met grote kans verschillend zijn, maar die wel zekere overeenkomsten met elkaar hebben. Een andere optie is door met koppelgewichten te werken, en de cut-offs lager in te stellen zodat er meer koppelbare records overblijven. Het is mogelijk dat hier een geheel ander oplossing uit komt dan we eerder hadden verkregen. Als dat niet gewenst is moeten we de resterende records apart van de wel gekoppelde records met elkaar zien te koppelen. Mogelijk betekent dit op zijn beurt weer dat er flink wat water in de wijn gedaan moet worden.

9.5 Koppelen van persoonsgegevens

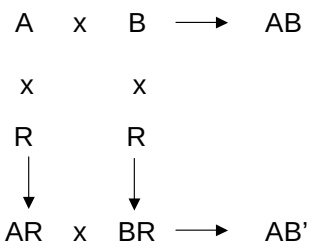
We bespreken hier een voorbeeld van een koppelsituatie en hoe die aan te pakken. Gegeven een register R met informatie over personen, waarin naast secundaire sleutels ook een primaire sleutel aanwezig is. Daarnaast zijn er twee bestanden A en B met persoonsgegevens. A en B bevatten geen primaire sleutel, maar wel secundaire sleutelvariabelen, zoals naam, adres, leeftijd etc. We veronderstellen:

- A en R zijn te koppelen op basis van gemeenschappelijke secundaire sleutelvariabelen
- B en R zijn te koppelen op basis van gemeenschappelijke secundaire sleutelvariabelen
- A en B zijn te koppelen op basis van gemeenschappelijke secundaire sleutelvariabelen

De gemeenschappelijk variabelen hoeven in de drie situaties niet dezelfde te zijn. Om A aan B te koppelen kunnen verschillende wegen bewandeld worden.

1. A wordt rechtstreeks aan B gekoppeld, gebruik makend van de gemeenschappelijke secundaire sleutelvariabelen in beide bestanden. Dit levert bestand AB op. In notatie: $A \times B \rightarrow AB$
2. A wordt aan R gekoppeld, en B wordt aan R gekoppeld, ieder op basis van gemeenschappelijke variabelen. De koppelbestanden AR en BR worden vervolgens gekoppeld op basis van de primaire sleutel uit R. Deze laatste koppeling is dus een join. Dit levert het koppelbestand AB' op. In notatie: $A \times R \rightarrow AR$, $B \times R \rightarrow BR$ en $AR \times BR \rightarrow AB'$.

Figuur 9.2: Koppelingsmogelijkheden van persoonsbestanden A en B: rechtstreeks of via een register R



Veronderstelling: al de getoonde koppelingen zijn mogelijk. Iedere koppeling heeft zijn eigen set koppelvariabelen. Koppeling AB x BR is een join. De andere koppelingen zijn koppelingen met secundaire sleutelvariabelen.

In figuur 9.2. staan deze koppelingen nog eens schematisch aangegeven. Omwille van het overzicht zijn de details over de afzonderlijke koppelingen weggelaten, in het bijzonder ten aanzien van de gebruikte koppelvariabelen.

A priori is niet aan te geven welk koppelbestand beter is AB of AB'. Het is wel interessant om in een experiment beide koppelmethodieken toe te passen en de resultaatbestanden te vergelijken. Welk van beide koppelmethoden in de praktijk het beste werkt is niet in het algemeen te zeggen. Welke überhaupt mogelijk zijn hangt af van de beschikbare gemeenschappelijke koppelvariabelen.

9.6 Koppelen van bedrijfsgegevens

9.6.1 Specifieke aspecten:

In de koppelliteratuur zijn de achtergronden en de voorbeelden veelal gebaseerd op het koppelen van persoonsgegevens. Hoewel er vergelijkbare problemen optreden bij het koppelen van bedrijfsgegevens zijn er ook duidelijke verschillen. Men dient bij het koppelen van bedrijfsgegevens onder meer rekening te houden met de volgende aspecten:

- Een bedrijf is een construct. Zo is het niet altijd duidelijk wat nu wel en wat niet tot een bedrijf behoort. De samenstelling en betekenis van bijvoorbeeld de (statistische) eenheid Onderneming bij het CBS is mogelijk anders dan de samenstelling en betekenis van een Onderneming bij de Belastingdienst of in de buitenwereld[23]. Dat kan betekenen dat men denkt twee dezelfde eenheden te koppelen, maar dit echter andere constructies van het bedrijf zijn;
- Op twee ijkmomenten in de tijd kunnen de identificerende kenmerken van een bedrijf in twee bestanden gelijk zijn, dus gekoppeld worden. Tussen de twee tijdstippen kunnen er echter allerlei events (bijvoorbeeld fusie of splitsing) hebben plaatsgevonden, waardoor de gegevens van hetzelfde bedrijf op die twee verschillende ijkmomenten niet onderling meer vergelijkbaar zijn. Daarmee is bijvoorbeeld het onderling vergelijken van variabelen uit de verschillende gekoppelde bestanden, bijvoorbeeld via kengetallen, niet meer zo evident. Dat ligt anders als men naar de ontwikkeling van het bedrijf in de tijd wil kijken door het opbouwen van een reeks. In dat geval kan juist het probleem zijn dat het nog wel om hetzelfde bedrijf gaat maar dat de identificerende variabelen, bijvoorbeeld de naam van de rechtspersoon of BEID, op de verschillende ijkmomenten niet meer aan elkaar gelijk zijn;
- De identificerende kenmerken op basis van de naam van hetzelfde bedrijf kunnen sterk verschillen. Te noemen zijn: de handelsnaam, naam van de eigenaar, naam van de rechtspersoon, naam van een onderdeel van het bedrijf (de productie-eenheid), een algemeen gehanteerde naam (bijvoorbeeld in de reclame) en de naam van de accountant of het servicekantoor. Dit geldt ook voor het adres. Daarbij kan het onder meer gaan om het postadres of het bezoekadres, het adres van het hoofdkantoor of het adres van de vestiging, of het adres van de accountant of het servicebureau, dat de administratie voert;
- De gegevens zoals verzameld bij de bedrijven zijn, veel meer dan bij personen, sterk onderling gerelateerd. Dat kan betekenen dat het berekenen van bijvoorbeeld kengetallen, zoals productiviteit, uit twee gekoppelde bestanden niet zonder gevaar is. Men moet er heel

[23] Bijvoorbeeld meer of minder zeggenschapsrelaties zijn meegenomen.

zeker van zijn dat het gaat om dezelfde constructie van het bedrijf en dat de gebruikte waarden nagenoeg op hetzelfde ijkmoment gemeten zijn;

- In één bedrijf kunnen meerdere economische activiteiten (zie SBI's) worden uitgevoerd. Bij het koppelen van bestanden kan dit leiden tot problemen, omdat niet altijd even duidelijk is welke activiteiten wel en welke niet zijn meegenomen in de verschillende gekoppelde bestanden. Denk bijvoorbeeld aan Pensioen BV's. Een ander gerelateerd probleem doet zich vaak voor bij functionele statistieken. Daar wil men de gegevens uit verschillende bestanden relateren aan die ene activiteit. Dat is veelal niet mogelijk, omdat bijvoorbeeld de omzet, kosten e.d. gaan over het gehele bedrijf en niet zijn onderverdeeld naar de aparte activiteiten;
- Omdat men te maken heeft met verschillende constructies van een bedrijf, is er vaak behoefte om gegevens van die constructies aan elkaar te relateren. Dit door eenheden, bijvoorbeeld de Bedrijfseenheden, bij elkaar op te tellen tot een grotere eenheid, bijvoorbeeld de Onderneming, en de totalen daarvan te koppelen aan andere bestanden met gegevens over die grotere eenheid. Dit is echter niet altijd evident en vergelijkbaar. Zo kan er bij de gegevens van de grotere eenheid sprake zijn van consolidatie, d.w.z. de onderlinge leveringen en stromen tussen de kleinere eenheden worden niet meegenomen. De optelsom van de kleinere eenheden hoeft niet altijd gelijk te zijn aan het geconsolideerde totaal;
- De administratieve verwerkingen van events en mutaties bij bedrijven verloopt vaak veel trager dan die bij mutaties in persoonsgegevens, waardoor bestanden nog "vervuild" kunnen zijn met oude gegevens. Daarbij kan er ook nog een verschil optreden tussen het tijdstip, waarop het event heeft plaatsgevonden, en het tijdstip, waarop dat administratief is vastgelegd en verwerkt in bestanden.

9.6.2 Belangrijkste (statistische) bedrijfseenheden:

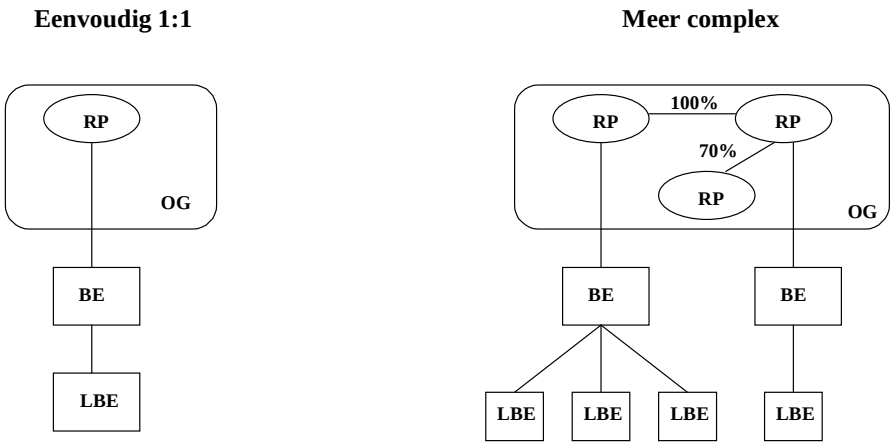
Bij het koppelen van bedrijfsgegevens heeft men te maken met meerdere gerelateerde constructies of eenheden, die op één of ander wijze een bedrijf kunnen vertegenwoordigen. Daarbij gaat het (vooral statistisch gezien) om de volgende eenheden: (1) Rechtspersoon (bij het CBS een CBS-persoon genoemd; dit kan zowel een rechtspersoon als een natuurlijk persoon zijn), een Onderneming, een Bedrijfseenheid en een Vestiging (Lokale Bedrijfseenheid). Grosso modo is de Onderneming (*OG*) de centrale eenheid. Een OG wordt samengesteld uit één of meer Rechtspersonen. Als het gaat om meer dan één Rechtspersoon dan dienen er zeggenschapsrelaties te zijn tussen die Rechtspersonen. Dat hoeft geen 100% zeggenschap te zijn. Die zeggenschapsrelaties zijn gebaseerd op informatie van de Kamer van Koophandel en de Belastingdienst. De OG is de financiële actor in het economische proces. De Rechtspersonen en dus de OG kunnen één of meer productie-eenheden, bij het CBS Bedrijfseenheden (*BE*) genoemd, vertegenwoordigen. Het gaat hier om de productieve actor in het economische proces. Bij het CBS zijn de meeste bedrijfsstatistieken gebaseerd op de BE. BE's kunnen meerdere vestigingen hebben op verschillende lokaties.

9.6.3 Ontwikkelingen

Er is bij onderzoekers steeds meer behoefte om bedrijven in de tijd te kunnen volgen (economische demografie). Daartoe dient een reeks te worden opgezet waarbij, via elk event, het duidelijk is

welke eenheden in een nieuwe situatie zijn ontstaan uit eenheden uit de oude situatie of eventueel nieuw zijn ontstaan of zijn beëindigd. Daarbij gaat het niet alleen om 1:n of n:1 relaties, maar ook om n:m relaties. Dit maakt het complex. Daarbij is het van belang om zowel het event als het tijdstip waarop het event heeft plaatsgevonden vast te leggen. De complexiteit neemt verder toe als niet alleen een reeks wordt opgezet, maar ook het aantal variabelen wordt uitgebreid waarop die reeks betrekking heeft. Hierdoor moet niet alleen “horizontaal” maar ook “verticaal” worden gekoppeld.

Figuur 9.3: Verschillende (statistische) eenheden bij bedrijfsstatistieken



RP = Rechtspersoon
OG = Ondernemingengroep
BE = Bedrijfseenheid
LBE = Vestiging

Een andere ontwikkeling is dat er in toenemende mate behoefte is om persoons- en bedrijfsgegevens te koppelen. Bijvoorbeeld om te kijken wat voor soort mensen waar werken. Dit soort koppelingen vraagt om het beschikbaar zijn van een te koppelen primaire sleutel in het bestand met de “grootste eenheid”, in dit geval het bedrijf, en een te koppelen foreign key in het bestand met de kleinste eenheid, in dit geval de werknemer.

Bij het CBS is onderscheid gemaakt tussen grote en complexe Ondernemingen, de zogenaamde Top X, en de rest. Bij de Top X Ondernemingen wordt nog onderscheid gemaakt tussen verschillende Bedrijfseenheden (BE’s) op basis van een aantal kenmerken, waaronder de homogeniteit van de activiteit. Dat geldt vanaf 1-1-2010 niet meer voor de rest van de Ondernemingen. Daar worden de verschillende activiteiten (als die er zijn) op één hoop gegooid. Daar geldt OG = BE. Achtergrond: een betere koppeling, via de koppeling met de OG, mogelijk te maken met gegevens uit externe bronnen. Deze koppeling met externe bronnen, en dan met name

die van de Belastingdienst, is al verbeterd door de afleiding van de OG beter te laten aansluiten bij de eenheden van de Belastingdienst.

10. Literatuur

De Jong, W.A.M. (1991), *Technieken voor het koppelen van bestanden*. Statistische onderzoekingen, M41, SDU/uitgeverij / CBS-publikaties, 's-Gravenhage.

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

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

Gartner (2007), *Magic quadrant for data quality tools 2007*. Gartner RAS, research note, juni 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. en 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. en Bibby, J. (1982), *Multivariate analysis*. Academic Press.

Nemhauser, G.L en 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. en James, A.P. (1959), Automatic linkage of vital records. *Science* 130, 954-959.

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

Sankoff, D. en 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*. Interne nota, CBS, Voorburg.

Wikipedia, artikel over het EM-algoritme, http://en.wikipedia.org/wiki/EM_algorithm.

Wikipedia, artikel over Record linkage, http://en.wikipedia.org/wiki/Record linkage.

Willenborg, L. en 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. Het Fellegi-Sunter-model

Het model van Fellegi en Sunter (1969) is gebaseerd op een beslissingstheoretische benadering, die een procedure afkomstig van Newcombe formaliseert (zie Newcombe e.a., 1959). Omdat de methode van Fellegi en Sunter van grote invloed is geweest in de wereld van het koppelen (en nog steeds is), beschrijven we die hier kort, aan de hand van De Jong (1991).

We gaan uit van twee verzamelingen met eenheden A en B, beide behorend tot een populatie P. Een typisch element van A is a, en van B is dat b. We veronderstellen dat er elementen zijn die in beide populaties voorkomen, dus dat $A \cap B \neq \phi$. De records corresponderend met de eenheden in A en B noteren we met $\alpha(a)$ voor $a \in A$ en met $\beta(b)$ voor $b \in B$. De bij A en B behorende bestanden noteren we met $\alpha(A)$, $\beta(B)$. Het is zaak de verzamelingen eenheden (A en B) te onderscheiden van hun representaties in de vorm van bestanden ($\alpha(A), \beta(B)$). Ook de representaties α en β dienen van elkaar onderscheiden te worden, omdat ze van elkaar kunnen verschillen. De representaties zijn inclusief fouten in de opgave, verwerkingsfouten, etc., kortom, alle niet-steekproeffouten.

Het kan bijvoorbeeld zijn dat de voornaam van eenzelfde persoon in het ene bestand als 'Hugo' (doopnaam) is weergegeven en in het andere als 'Huug' (roepnaam). Dit is dus een voorbeeld van een eenheid a met $\alpha(a) \neq \beta(a)$. Evenzo kan het voorkomen dat er twee verschillende eenheden a, b zijn (dus $a \neq b$) met $\alpha(a) = \beta(b)$.

We beschouwen nu twee belangrijke deelverzamelingen van de verzameling van recordparen uit $\alpha(A)$ en $\beta(B)$, namelijk $\alpha(A) \times \beta(B) = \{(\alpha(a), \beta(b) \mid a \in A, b \in B\}$:

1. de *matchverzameling*: $M = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), a = b\}$.
2. de *unmatchverzameling*: $U = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), b \in \beta(B), a \neq b\}$.

Deze verzamelingen M en U zijn in de praktijk niet bekend

De vergelijkingsvector $\gamma = (\gamma_1, ..., \gamma_K)$ geassocieerd met de records in beide bestanden is als volgt gedefinieerd:

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

waarbij iedere γ_i, $i = 1, ..., K$ een specifieke vergelijking symboliseert. Zo kan γ_1 een indicator zijn die overeenstemming in het geslacht van twee personen registreert (wel/niet van hetzelfde geslacht). γ_2 zou een indicator kunnen zijn of twee familienamen wel/niet hetzelfde zijn. Enzovoort. Zij $\Gamma = \{0,1\} \times ... \times \{0,1\}$ (K keer), de verzameling mogelijke realisaties van γ. Op basis van γ moet een recordpaar worden geclassificeerd als behorend tot M of tot U. Twee records worden gekoppeld als het recordpaar geclassificeerd is als behorende tot M. De verzameling van recordparen $\alpha(A) \times \beta(B)$ valt daardoor uiteen in twee verzamelingen

3. de *koppelverzameling* (verzameling 'links'):
   $L = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), \alpha(a)$ en $\beta(b)$ zijn gekoppeld$\}$.

4. de *verzameling niet gekoppelde records* (‘non-links’):

$N = \{(\alpha(a), \beta(b)) \mid \alpha(a) \in \alpha(A), \beta(b) \in \beta(B), \alpha(a)$ en $\beta(b)$ zijn niet gekoppeld$\}$

Bij het koppelen wordt er naar gestreefd L zoveel mogelijk op M te laten lijken (of, equivalent, N op U). Dit komt neer op het streven om miskoppelingen en gemiste koppelingen te vermijden. Er geldt:

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

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

Fellegi en Sunter (1969) nemen als koppelcriterium de verhouding

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

waar

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

de fractie recordparen is $\alpha(A) \times \beta(B)$ in M met scorevector γ, en

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

de fractie recordparen $\alpha(A) \times \beta(B)$ in U met score-vector γ. Een praktisch probleem is dat $m(\gamma)$ en $u(\gamma)$ niet bekend zijn. Men kan ze in eerste instantie echter benaderen.

Idealiter zou men één parameter c willen kunnen instellen zodanig dat

Als $R(\gamma) \geq c$ dan worden $\alpha(a)$ en $\beta(b)$ gekoppeld,
Als $R(\gamma) < c$ dan worden $\alpha(a)$ en $\beta(b)$ niet gekoppeld,

In de praktijk blijkt één parameter vaak niet mogelijk te zijn om M en U te scheiden. Daarom werkt men daar liever met twee cut-off-waarden $c \leq d$ zodanig dat

Als $R(\gamma) > d$ dan worden $\alpha(a)$ en $\beta(b)$ gekoppeld,
Als $R(\gamma) < c$ dan worden $\alpha(a)$ en $\beta(b)$ niet gekoppeld,
Als $c < R(\gamma) < d$ dan worden $\alpha(a)$ en $\beta(b)$ als voorlopige koppelingen beschouwd

Een speciaal computerprogramma is nodig om aan de hand van een γ de recordparen te elimineren waarvoor $R(\gamma) < c$ en de recordparen op te sporen waarvoor $c < R(\gamma) < d$ geldt, die vervolgens door een koppelspecialist nader bekeken moeten worden om na te gaan welke daarvan koppelingen zijn en welke niet. Deze inspectie is zeer tijdrovend en kostbaar en men zou die tot een minimum willen beperken. De Fellegi-Sunter-methode heeft juist dit als doel.

We stoppen hier de bespreking van de methode die door Fellegi en Sunter is ontwikkeld. Het basis-idee van hun aanpak moge nu duidelijk zijn. Het moge ook duidelijk zijn dat nog allerlei schattingen moeten worden verricht om de methode in de praktijk ook daadwerkelijk te kunnen toepassen. Een probleem is dat de verzamelingen M en U niet bekend zijn, en dus ook niet $R(\gamma)$ voor een gegeven γ. Voor dit soort zaken verwijzen we de geïnteresseerde lezer naar het oorspronkelijke artikel van Fellegi en Sunter, naar De Jong (1991) of naar Herzog e.a. (2007, hoofdstuk 9).

Appendix B. Meer over metrieken

Hier willen we de discussie voortzetten die aan het eind van paragraaf 7.3.1.1 is gestopt. Het betreft een uitweiding die voor de praktijk weliswaar van belang is, maar nogal specialistisch en technisch is.

Om de draad van paragraaf 7.3.1.1 weer op te pakken: we bekijken het probleem om tekst, i.c. namen, die verkeerd kunnen zijn gespeeld, omdat ze bijvoorbeeld zijn opgeschreven louter op basis van de uitspraak. Het is dan onmogelijk om te weten of iemand 'Jansen', 'Janssen', 'Janszen', etc. heet.[24] We zouden dergelijke namen graag met elkaar in verband willen brengen door aan namen een fonologisch-geïnspireerde code te koppelen.

We hebben in dit geval dus een verzameling N (zeg van namen met allerlei spellingsvarianten) en een verzameling P codes gebaseerd op de uitspraak (zoals Soundex)[25] en een functie $f : N \to P$ die we surjectief kunnen veronderstellen.[26] Deze f introduceert een equivalentierelatie $\approx$ op N, met $n_1 \approx n_2$ voor $n_1, n_2 \in N$ als $f(n_1) = f(n_2)$. Via deze f kunnen we een metriek d_L^* op P introduceren, afgeleid van d_L, de metriek op N, als volgt: voor $a, b \in P$ zijn $f^{-1}(a) \subseteq N$ en $f^{-1}(b) \subseteq N$ de niet lege (f is surjectief!) volledige originelen van a en b, met $f^{-1}(a) \cap f^{-1}(b) = \phi$ als $a \neq b$. Dan is

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

waar de laatste gelijkheid de zogenaamde Hausdorff-afstand d_{Haus} gebaseerd op d, voorstelt. Hier stellen 'sup' en 'inf' 'supremum', respectievelijk 'infimum' voor, hetgeen we in onze context gevoeglijk kunnen vervangen door maximum en minimum (omdat we in de praktijk met eindige sets (woorden, namen, etc.) te maken hebben). We vinden derhalve:

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

Dit is een metriek op 2^N, de collectie deelverzamelingen van N, als tenminste d begrensd is, dat wil zeggen $d(x,y) \leq M$ voor alle $x, y \in N$ voor een zekere $M > 0$[27]. Deze metriek is als volgt gedefinieerd voor verzamelingen $A, B \subseteq N$:

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

---

[24] Verificatie bij de naamdrager zou dit probleem kunnen vermijden. Maar we veronderstellen even dat dat niet het geval is. Bij eigennamen ligt verificatie voor de hand. Maar bij een tekst die, zeg, iemands beroep omschrijft is een interviewer waarschijnlijk niet zo snel geneigd om zijn eigen spel(on)vermogen te etaleren.

[25] Op basis van een setje regels wordt uit een gegeven string dan een code afgeleid. In de bijdrage over coderen in de Methodenreeks zijn de regels voor de Nederlandse variant van Soundex gegeven.

[26] Zo niet, dan beperken we het bereik (codomain) van f tot $f(P) \subseteq P$. $f : P \to f(P)$ is surjectief.

[27] Anders is het geen metriek maar een 'extended metric', die ook de waarde ∞ kan aannemen.

Idealiter zouden alle spellingsvarianten van een woord in dezelfde equivalentieklasse terecht moeten komen en dan de afstand 0 tot elkaar moeten hebben. Daarbij zou men bijvoorbeeld gebruik kunnen maken van het Soundex-algoritme (voor Nederlands).

In Herzog e.a. (2007, hoofdstuk 13) worden 'comparator metrics' ingevoerd die bedoeld zijn voor strings met typografische fouten, zoals transposities (verwisseling van twee opeenvolgende karakters, zoals 'timmeramn' in plaats van 'timmerman'). De ene is afkomstig van Jaro en de andere van Winkler. Zonder verdere motivatie geven we ze hieronder. Voor meer informatie zie Winkler e.a. (2007). De Jaro similariteitsmaat is als volgt voor twee strings s en t:

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

waar:

- w_s is het gewicht geassocieerd met de s, w_t is het gewicht geassocieerd met de t, $w_{transpos}$ is het gewicht geassocieerd met de transpositie, met $w_s, w_t, w_{transpos} > 0$ en $w_s + w_t + w_{transpos} = 1$.
- $c > 0$ is het aantal karakters dat s en t gemeen hebben, waarbij gemeenschappelijke karakters minder dan de lengte van de kortste string (dus $\min\{l_s, l_t\}$) van elkaar verwijderd zijn.
- l_s is de lengte van s, l_t is de lengte van t
- τ is het aantal getransponeerde karakters

Als $c = 0$ dan $d_{Jaro}(s,t) = 0.$

De Winkler-similariteitsmaat is als volgt gedefinieerd:

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

waar $i = \min\{j,4\}$, met j het aantal karakters dat de strings s en t aan het begin gemeen hebben.

Een andere manier om strings te vergelijken is door ze op te delen in trigrammen en te tellen hoeveel trigrammen met elkaar matchen, en in welke mate. We handhaven hierbij de volgorde van de trigrammen in de string. Ook bij het matchen handhaven we de volgorde. Bij het matchen tellen we het aantal karakters die overeenkomen en op dezelfde positie staan en het aantal karakters dat hetzelfde is, ongeacht hun plaats in het trigram.

Appendix C. Overwegingen bij de selectie van koppelsoftware

Algemeen:

1. Gaat het om generieke software of is het alleen gericht op specifieke toepassingen (bijvoorbeeld alleen het koppelen op naam of alleen voor persoonsgegevens)?
2. Is de verkoper een betrouwbare en solide onderneming? Kan het technische support leveren? Wat is de visie van de verkoper op de langere termijn?
3. Gaat het om een compleet systeem (koppelen “out of the box”) of een set van componenten, waarom heen nog een systeem moet worden gebouwd? Hoe compleet is dan die set componenten?
4. Hoe goed is het systeem gedocumenteerd? Kan de gebruiker gaande weg het pakket zelf leren of heeft hij/zij (veel) ondersteuning nodig? Wordt er training geleverd?
5. Is er een gebruikersgroep?
6. Hoe goed sluit het pakket aan op de andere software en databases die het CBS zelf heeft?
7. Wat voor soort koppelingen staat het toe, bijvoorbeeld één file met zich zelf (ontdubbelen), van twee files, meer dan twee files, inclusief het koppelen met referentiefiles?
8. Hoe snel kan een gemiddeld koppelproject het systeem implementeren?
9. Kan het pakket en het koppelproces (gemakkelijk) worden aangepast aan de specifieke wensen van de gebruiker?

Kosten:

1. Wat zijn de aanschaf en onderhoudkosten (bijvoorbeeld aansluiting op andere software of database en upgrades) van het pakket?
2. Hoeveel kost een (site-)licentie?
3. Hoeveel kost het om mensen op te leiden?

Pre-verwerkingsfase:

1. Welke mogelijkheden zijn er voor de pre-verwerkingsfase? Zijn er mogelijkheden om (groepen van of gerelateerde) variabelen te controleren, te editen en te standaardiseren? In hoeverre kan de gebruiker zelf acties op variabelen en records definiëren? Kunnen deze acties ook worden uitgevoerd op groepen records (onder zelf te definiëren condities)?
2. Is het mogelijk om bijvoorbeeld postcodes “uit elkaar” te halen?
3. Leiden de bewerkingen tot een nieuwe file (de oude waarden zijn niet meer beschikbaar) of worden extra variabelen toegevoegd?
4. Zijn er mogelijkheden om te ontdubbelen?
5. Kan er gewerkt worden met blocking variabelen? Is het per run mogelijk om meer dan één blocking variabele (en mogelijke verschillende vergelijkingsmethoden) te definiëren?
6. Kan de gebruiker subsets van de file definiëren op basis waarvan gekoppeld moet worden? Kunnen (groepen van) records (tijdelijk) opzij worden gezet?
7. Ondersteunt het pakket het selecteren van kleinere “proefbestanden” om een testrun uit te voeren?
8. Zijn er mogelijkheden om “commentaarvelden” (vooraf en achteraf) te definiëren?;

Koppelmethoden:

1. Welke koppelmethoden worden ondersteund?
2. Kan zowel worden gekoppeld op personen als op bedrijven of andere identifiers?

3. Is het koppelproces een black box of kan de gebruiker het sturen met behulp van parameters? Zo ja, kan een file met parameters gemakkelijk worden aangemaakt?
4. Hoeveel variabelen in de koppelsleutel zijn toegestaan? Kan het systeem ook overweg met foreign key koppelingen?
5. Is het pakket geschikt voor koppelingen op basis van gerelateerde eenheden (“is ontstaan uit” en “komt voort uit”)?
6. Kan de gebruiker de koppelvariabelen en de gewenste vergelijkingen definiëren?
7. Wat voor vergelijkingen van (de variabelen van) de sleutels zijn mogelijk? Bijvoorbeeld: karakter-voor-karakter, Soundex (fonetisch), string-vergelijkingen, metrieken, vergelijkingen van postcodes (met numeriek en alfanumeriek deel), datum en tijd vergelijkingen, vergelijkingen op basis van condities, die door de gebruiker kunnen worden bepaald?
8. Kan er gebruik worden gemaakt van metrieken, kansen, cut-off-waarden (helpt het systeem met het doen van suggesties om de cut-off-waarde vast te stellen of levert het daarvoor informatie)?
9. Hoe gaat het pakket om met missings in de sleutelvariabelen (bijvoorbeeld een gewicht 0), ook als er sprake is van een methode met gewichten?
10. Kan de gebruiker bijvoorbeeld kritieke variabelen aangeven, waarvan hij/zij heeft vastgesteld dat records alleen koppelen als deze variabelen (van de sleutel) overeenkomen? Kan er bij de vergelijking per variabele worden gedifferentieerd?
11. Kunnen per variabele en voor het totaal gewichten worden berekend en welke methoden zijn daarvoor beschikbaar?
12. Kan bij het koppelen ook rekening gehouden worden met de afhankelijkheden tussen variabelen?

Post-verwerkingsfase:

1. Ondersteunt het pakket mogelijkheden om schattingen te maken van de eerste en tweede orde fouten (kwaliteit van de koppelingen)?
2. Welke mogelijkheden zijn er om overzichten te genereren, bijvoorbeeld om een evaluatie uit te voeren (met bijvoorbeeld gekoppelde en niet gekoppelde records, twijfelgevallen, berekende gewichten e.d.)? Kan het formaat en de inhoud van de overzichten worden aangepast? Zijn er ook overzichten in grafische vorm?
3. Zijn er ook mogelijkheden om statistieken te genereren om het proces te evalueren?
4. Kunnen de resultaten van meerdere verschillende runs gemakkelijk met elkaar worden vergeleken?
5. In welke formaten kunnen de resultaten van de koppeling worden weggeschreven en opgeslagen?

Data- en systeembeheer:

1. Op welk platform kan de software gebruikt worden? En welke hardware eisen worden gesteld?
2. Is het een single-user of multi-user pakket?
3. Kan het systeem interactief werken of alleen in de batch?
4. Welk dataformaat en -opslag kan het pakket aan? Past dat formaat in de bestaande informatie-architectuur?
5. Wat is de maximum omvang van de files (aantal records) dat het pakket kan verwerken, inclusief de resulterende file met koppelkandidaten?
6. Hoe verwerkt het pakket de records (bijvoorbeeld tijdelijke file, gesorteerd of op basis van pointers)?

7. Ondersteunt het pakket functies om de data (in de tussenstappen of alleen aan het eind van het proces) te bekijken en te manipuleren?
8. Hoe ziet de interface van het pakket eruit?
9. Hoe goed “performed” de software, met name als het gaat om grote files en meer geavanceerde methoden?