FACULTEIT
INGENIEURSWETENSCHAPPEN

DEPARTEMENT
ELEKTROTECHNIEK – ESAT

KATHOLIEKE
UNIVERSITEIT
LEUVEN

Zelflerende Spraakherkenning via Matrix-factorisatie

Eindwerk voorgedragen tot het behalen van het diploma van Burgerlijk elektrotechnisch ingenieur, optie Multimedia & Signaalverwerking

Alexander Bertrand

Promotor:
Prof. Dr. Ir. H. Van hamme

Dagelijkse begeleiding:
Dr. Ir. Veronique Stouten
Dr. Ir. Kris Demuynck

2006 – 2007

Woord vooraf

Toen ik vorig jaar op zoek was naar een interessant thesis-onderwerp, was mijn belangrijkste voorwaarde dat het onderwerp nog niet beschreven zou zijn in de literatuur. Uiteraard houdt dit een zeker risico in. Het feit dat het onderwerp dat ik koos op data-mining technieken gebaseerd is, betekent bovendien dat de resultaten heel onvoorspelbaar zijn. Data-mining wordt namelijk vaak geassocieerd met 'black-box' technieken. Ik had echter het gevoel dat mijn begeleiders Kris en Veronique er vertrouwen in hadden dat de matrix-factorisatie algoritmes die ik zou gebruiken wel degelijk in staat waren om nuttige resultaten te leveren. Dit laatste heeft, samen met de enthousiaste uitleg van Prof. Van Hamme op de thesis-beurs, mijn keuze bepaald.

Het onderzoek zelf kende hoogtes en laagtes. Mijn ideeën, en de implementatie ervan, resulteerden vaak in een ontgoocheling na het aanschouwen van de resultaten. Het trial & error karakter van de experimenten was vaak frustrerend. Bovendien was de rekentijd voor de meeste experimenten heel lang, zodat ik meestal een à twee dagen moest wachten op de resultaten.

Dit alles wordt echter ruimschoots gecompenseerd door mijn interesse in het onderwerp en bepaalde experimenten die toch tot een interessant resultaat leidden. Hoewel de resultaten geen revolutie zullen veroorzaken in de wereld van de spraakherkenning, denk ik toch dat dit eindwerk een interessante wetenschappelijke waarde heeft. Hierbij denk ik vooral aan de resultaten van het deelprobleem omtrent kenmerken-extractie (zie tekst).

Ik wil mijn begeleiders Kris en Veronique bedanken voor hun uitstekende begeleiding. Zij hebben mij verschillende goede ideeën aangereikt. Bovendien waren zij een belangrijke hulp bij praktische zaken, zoals bv. het gebruik van SSH, Linux en de ESAT-spraakherkenner. Ook de vele non-MATLAB scripts van Kris waren een noodzaak bij mijn experimenten, waarvoor ik hem heel dankbaar ben. Veronique wil ik ook nog eens extra bedanken voor de belangrijke opmerkingen en verbeteringen bij het doornemen van mijn tekst, wat de kwaliteit van dit eindwerk zeker ten goede is gekomen. Ook Joram Bekaert en mijn broer Jan Bertrand ben ik heel dankbaar voor de feedback na het doornemen van mijn eindwerk. Tot slot gaat mijn dank uit naar mijn promotor Prof. Van Hamme voor zijn vertrouwen, de crash-courses en de nuttige feedback tijdens de vergaderingen.

Het staat vast dat er nog heel wat onderzoek kan gebeuren op het domein van de zelflerende spraakherkenning. Ik hoop dat mijn opvolgers met evenveel interesse en voldoening aan dit onderwerp kunnen verderwerken. Ik wens ze alvast veel succes toe.

Alexander Bertrand

Leuven, 15 mei 2007

Abstract

De mens slaagt erin om gedurende de eerste levensjaren spraak te leren begrijpen, zonder dat daarvoor kennis over fonemen, woorden of zinnen beschikbaar is. Dit suggereert dat het menselijk brein in staat is om de impliciete structuur in spraakdata terug te vinden door enkel gebruik te maken van de akoestische signalen. Dit staat in schril contrast met state-of-the-art spraakherkenningssystemen. Om deze systemen te trainen is er nood aan een manuele transcriptie van de data. Het systeem moet voor elk tijdstip in het spraaksignaal weten met welk foneem het spraaksegment overeenkomt. Het moet ook kennis hebben omtrent de woorden die worden uitgesproken en omtrent de foneeminhoud van deze woorden.

Het doel van dit eindwerk is om na te gaan of een computer in staat is om zelf de latente structuur in spraaksignalen te vinden zonder dat hiervoor a-priorische kennis gebruikt wordt. Dit gebeurt aan de hand van matrix-factorisatie technieken, die in staat zijn om bij hoge dimensionaliteit structuur te ontdekken in data. Niet-negatieve matrix-factorisaties genieten hierbij de voorkeur t.o.v. factorisaties zonder niet-negativiteitsvoorwaarden, omdat deze algoritmes structuren vinden die over het algemeen dichter aanleunen bij menselijke perceptie. In dit eindwerk ligt de focus op twee deelproblemen op het laagste niveau van het spraakherkenningssysteem: kenmerken-extractie en foneemclassificatie.

In het eerste deelprobleem wordt gezocht naar een nieuwe kenmerkenset ter vervanging van de MEL-filterbank die in veel spraakherkenningssystemen gebruikt wordt als kenmerken-extractor. Hiervoor worden drie verschillende factorisatie-technieken getest. De eerste is de singuliere-waardenontbinding, die een factorisatie vindt die sterk gelijkt op een discrete cosinustransformatie. De tweede techniek is een niet-negatieve matrix-factorisatie (NMF) op basis van de gemiddelde kwadratische fout op de reconstructie. Deze methode vindt een bruikbare kenmerkenset, maar slaagt er niet in de herkenningsresultaten te verbeteren. NMF volgens een divergentie-criterium tenslotte, vindt een factorisatie die heel sterk op de MEL-filterbank gelijkt. Er worden dan ook gelijkaardige herkenningsresultaten bereikt als bij het gebruik van de MEL-filterbank. Het feit dat de gevonden oplossing gelijkaardig is aan de MEL-filterbank, die op basis van het menselijk gehoorsysteem werd ontworpen, geeft aan dat spraakproductie en het gehoorsysteem goed op elkaar afgestemd zijn.

In het tweede deelprobleem wordt gepoogd om spraaksegmenten te classificeren in foneemklassen, zonder gebruik te maken van a-priorische foneemkennis. Dit gebeurt aan de hand van de factorisatie van een KNN-matrix (‘k *nearest neighbours*’), die op basis van de kenmerkenvectoren wordt opgesteld. De resultaten van NMF worden vergeleken met die van een methode gebaseerd op de eigenwaardenontbinding van de KNN-matrix. Geen van beide methodes slaagt erin om de juiste foneemclassificatie te genereren. Toch vinden beide methodes gelijkaardige overeenkomsten tussen bepaalde foneemklassen. Dit wijst erop dat er in de kenmerken-ruimte vermoedelijk een grote overlap is tussen deze foneemklassen, waardoor beide methodes falen om een correcte foneemclassificatie te vinden.

Lijst van symbolen en afkortingen

DCT	Discrete Cosinus Transformatie
HMM	Hidden Markov Model (verborgen Markov-keten)
SWO	Singuliere-waardenontbinding
LSA	Latent Semantische Analyse
PLSA	Probabilistische Latent Semantische Analyse
EM	Expectation Maximization
NMF	Niet-negatieve Matrix Factorisatie
MSE	Mean Squared Error (gemiddelde kwadratische fout)
$Div(\mathbf{V}\|\|\mathbf{X})$	Divergentie tussen matrix $\mathbf{V}$ en $\mathbf{X}$ (zie formule (3.17))
MFCC	MEL-Frequentie Cepstrale Coëfficiënten
Env	Omhullende-operator gedefinieerd in sectie 4.2.1
$Smooth$	Smoothness-operator gedefinieerd in sectie 4.2.1
PER	Phoneme Error Rate
IDCT	Inverse Discrete Cosinus Transformatie
PCA	Principale Componenten Analyse
KNN	k Nearest Neighbours
MLP	Multi-Layer-Perceptron
$\Phi(i)$	De verzameling van frame-indices die bij foneem i behoren volgens de manuele foneemtranscriptie van de TIMIT-databank
$Sp(\mathbf{x})$	'Spaarsheid' (*sparseness*) van de vector $\mathbf{x}$ (zie formule (5.8))

Inhoudsopgave

Lijst van figuren

Hoofdstuk 1

Inleiding

1.1 Context

Het is opmerkelijk dat de mens automatisch de klankpatronen van menselijke spraak kan leren tijdens de eerste levensjaren. Bovendien doet hij dit beduidend beter dan een spraakherkenningssysteem, hoewel er in deze laatste manueel expertkennis wordt ingebracht zoals bv. informatie omtrent fonemen[1]. Men zou dus kunnen stellen dat spraaksignalen een verborgen structuur bevatten die het menselijk brein zonder enige voorkennis zelf kan ontdekken. Hoe dit gebeurt is tot op heden nog steeds een raadsel.

Het onderliggende doel van de experimenten in dit eindwerk is om na te gaan of een computer, net zoals de mens, in staat is om de onderliggende structuur in spraaksignalen te ontdekken. In dit eindwerk ligt de focus op het laagste niveau van het spraakherkenningssysteem. Er worden twee deelproblemen beschouwd: kenmerken-extractie en foneemclassificatie. Net zoals een baby mag de computer hierbij enkel gebruik maken van een hele grote hoeveelheid continue spraak, zonder over enige informatie te beschikken omtrent welk foneem of welk woord er wordt uitgesproken. Dit laatste is een belangrijke nevenvoorwaarde die in alle experimenten wordt gehanteerd[2].

Het is echter niet de bedoeling om te achterhalen hoe een baby deze structuur ontdekt en welke structuren of patronen de mens gebruikt om spraak te begrijpen. Daarvoor bestaat nog te weinig kennis omtrent de werking van de hersenen. Een baby krijgt bovendien heel wat feedback op allerlei vlakken, wat in het geval van een computer ontbreekt. Het feit dat een baby op basis van voldoende voorbeelden spraak leert begrijpen, geeft wel aan dat er een zekere structuur in de spraaksignalen zelf verborgen zit. Er wordt dus enkel nagegaan of een computer ook dergelijke structuren kan blootleggen.

De laatste jaren zijn er heel wat data-mining technieken ontwikkeld waarmee het zoeken naar latente structuren in data door een computer kan gebeuren. Matrix-factorisatie technieken hebben reeds in meerdere domeinen aangetoond dat ze robuust werken en in staat zijn bij hoge dimensionaliteit structuren te ontdekken in data. Zo werden de singuliere-waardenontbinding en recent ook positieve en

[1] Fonemen zijn betekenisonderscheidende klankeenheden waaruit woorden worden opgebouwd (de volledige set fonemen is taalafhankelijk).

[2] Indien een computer zelfstandig een structuur moet zoeken zonder menselijke hulp wordt dit ook wel ‘unsupervised learning’ genoemd. Dit is het streefdoel van dit eindwerk. Indien er toch gebruik gemaakt wordt van menselijke hulp, zoals bv. een manuele classificatie van de data, valt dit onder de noemer ‘supervised learning’.

probabilistische matrix-factorisatie algoritmes succesvol toegepast in het domein van spraakherkenning om semantische relaties te modelleren tussen woorden (latent semantische analyse).

Matrix-factorisatie algoritmes ontbinden een matrix in een product van twee kleinere matrices. Op die manier wordt in de praktijk een compacte representatie bekomen van een grote observatiematrix. Het vinden van een factorisatie die zowel compact als nauwkeurig is - en dus weinig afwijkt van de oorspronkelijke matrix - impliceert bijna automatisch dat de latente structuur van het probleem is blootgelegd.

De experimenten in dit eindwerk steunen allemaal op dergelijke factorisatie-technieken. Er wordt hierbij een voorkeur gegeven aan niet-negatieve matrix-factorisaties omdat deze een deel-gebaseerde structuur kunnen vinden. In de literatuur wordt geargumenteerd dat de patronen die gevonden worden via niet-negatieve matrix-factorisatie dichter aanleunen bij menselijke perceptie dan patronen die via matrix-factorisatietechnieken zonder niet-negativiteitsvoorwaarden gevonden worden.

1.2 Overzicht

In hoofdstuk 2 wordt een korte introductie gegeven over spraakherkenning en de voorverwerking van de bemonsterde spraaksignalen in een spraakherkenningssysteem. Dit laat ons toe bepaalde begrippen te introduceren waarnaar in volgende hoofdstukken nog verwezen zal worden. In de tweede sectie van hoofdstuk 2 wordt dieper ingegaan op de MEL-filterbank, die in spraakherkenning vaak wordt gebruikt voor kenmerken-extractie.

Hoofdstuk 3 beschrijft de verschillende matrix-factorisatie algoritmes die werden gebruikt in de experimenten in dit eindwerk. De voor- en nadelen van elke methode worden besproken. Voor een beter begrip van de mogelijkheden en beperkingen van elk van deze technieken, worden ze uitgelegd aan de hand van een toepassing die ook in spraakherkenning aan belang wint: latent semantische analyse.

In hoofdstuk 4 wordt het eerste deelprobleem binnen dit eindwerk behandeld. Het doel is om een nieuwe kenmerkenset te vinden via matrix-factorisatie, die als alternatief zou kunnen dienen voor de MEL-filterbank om spraaksignalen compact te beschrijven.

Hoofdstuk 5 behandelt het tweede deelprobleem. Hier is het de bedoeling om een beter inzicht te verwerven in de akoestische ruimte van afzonderlijke spraakklanken. Via matrix-factorisatie wordt gepoogd de verschillende klassen in deze ruimte te onderscheiden in de hoop om op deze manier een foneem-model te verkrijgen.

Hoofdstuk 2

Basistechnieken in spraakherkenningssystemen

2.1 Inleiding

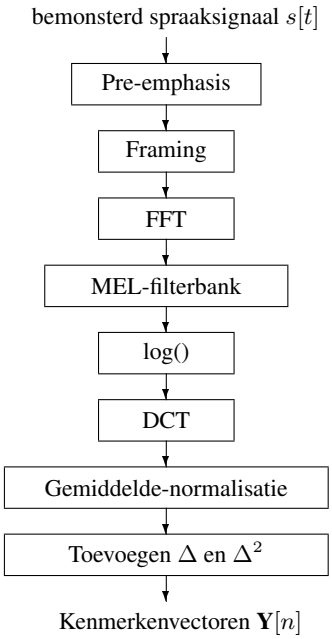
In de eerste sectie van dit hoofdstuk wordt kort ingegaan op de werking van een spraakherkenningssysteem. Er wordt een beknopte beschrijving gegeven van de basistechnieken die worden toegepast in een spraakherkenner. State-of-the-art herkenners omvatten nog heel wat aspecten die hier niet besproken zullen worden. Het is enkel de bedoeling om het onderwerp van dit eindwerk te situeren en de lezer vertrouwd te maken met enkele basisbegrippen die in de tekst aan bod zullen komen. Voor een meer complete beschrijving wordt verwezen naar de literatuur[1]. Bestaande spraakherkenningssystemen kunnen afwijken van het systeem dat in dit hoofdstuk wordt beschreven.

In de tweede sectie wordt dieper ingegaan op de MEL-filterbank die vaak wordt gebruikt voor kenmerken-extractie in de voorverwerking van een spraakherkenningssysteem. Het zoeken naar een geschikt alternatief voor kenmerken-extractie is een belangrijk onderdeel van dit eindwerk (zie hoofdstuk 4).

2.2 Werking van een spraakherkenningssysteem

In deze sectie wordt kort uitgelegd hoe een spraakherkenningssysteem werkt. Het eerste deel beschrijft hoe kenmerkenvectoren worden berekend in de voorverwerking die voorafgaat aan de eigenlijke spraakherkenning. De tweede paragraaf legt kort uit hoe de herkenning zelf gebeurt.

[1] Een duidelijke en compacte beschrijving van spraakherkenners met groot vocabularium is te vinden in [1]. Deze paper bevat een kort, maar duidelijk overzicht omtrent de algoritmes en modellen die in huidige spraakherkenners worden gebruikt. Voor een meer uitgebreide uitleg wordt verwezen naar [2].



Figuur 2.1: Voorverwerking van een spraakherkenningssysteem

2.2.1 Voorverwerking

Figuur 2.1 geeft de verschillende stappen weer die doorlopen worden tijdens de voorverwerking in een spraakherkenner. Het systeem krijgt een bemonsterd spraaksignaal aan de ingang. De uitgang bestaat uit een sequentie van kenmerkenvectoren die gebruikt kunnen worden voor de herkenning van spraak (zie sectie 2.2.2). Hieronder worden de verschillende stappen toegelicht:

1. **Pre-emphasis**: De amplitude van het spectrum van een spraaksignaal is dalend met een helling van ongeveer 6dB per octaaf. Dit wordt gecompenseerd aan de hand van een hoogdoorlaatfilter, die voor een versterking zorgt van de hogere frequenties.
2. **Opdeling in frames**: Het bemonsterde spraaksignaal wordt opgedeeld in overlappende frames (meestal met een lengte van ongeveer 30 ms). Binnen een frame kan verondersteld worden dat de frequentie-inhoud van het spraaksignaal stationair is. Dit is het gevolg van de mechanische traagheid van het biologisch systeem waarmee spraak wordt geproduceerd. Elk frame wordt gewogen met een venster om artefacten in het spectrum te vermijden als gevolg van de truncatie van het signaal.
3. **FFT**: Op elk frame wordt een Fast Fourier Transform uitgevoerd om de spectrale inhoud van het frame te verkrijgen. Wegens symmetrie wordt slechts de helft van de punten behouden.
4. **MEL-integratie**: Het spectrum wordt binnen een aantal driehoekige gewichtsvensters geïntegreerd. Zo worden een aantal MEL-coëfficiënten bekomen (typisch 20 à 30). Zie sectie 2.3 voor meer informatie over de MEL-filterbank.
5. **Log**: Van elke MEL-coëfficiënt wordt de logaritme genomen. Dit zorgt ervoor dat het dynamisch bereik van de coëfficiënten gecomprimeerd wordt. Hierdoor wordt de verdeling van de energie in de kenmerkenvectoren min of meer Gaussiaans.

6. **DCT**: Er wordt een discrete cosinus transformatie uitgevoerd op de bekomen kenmerkenvector. Dit zorgt voor een betere decorrelatie ('lineaire onafhankelijkheid' of 'lineaire scheidbaarheid') van de verschillende coëfficiënten in de kenmerkenvector. De DCT zorgt bovendien voor een grotere 'energiecompactheid'. Op deze manier kan de vector uit de vorige stap gereduceerd worden tot een kleinere vector (typisch 12 coëfficiënten), zonder dat hierdoor veel informatie verloren gaat[2].
7. **Gemiddelde-normalisatie**: Er wordt op elke kenmerkenvector een gemiddelde-normalisatie uitgevoerd. Dit zorgt ervoor dat de kenmerkenvectoren minder afhankelijk zijn van microfoonvolume, kamerakoestiek, etc.
8. **Toevoegen van Δ en $\Delta\Delta$**: De kenmerkenvector wordt aangevuld met de eerste en tweede afgeleide van elke coëfficiënt (op basis van opeenvolgende frames). Deze beschrijven de 'snelheid' en de 'versnelling' van de opeenvolgende kenmerkenvectoren. Dit zorgt ervoor dat ook tijdsinformatie wordt opgenomen in de kenmerkenvector. De lengte van de vector wordt in deze stap verdrievoudigd. Er is empirisch aangetoond dat het toevoegen van eerste en tweede afgeleiden aan de kenmerkenvector een significant positief effect heeft op de herkenning van spraak.

2.2.2 Verborgen Markov-ketens

Vanuit de bekomen kenmerkenvectoren (observatievectoren) uit de vorige sectie moet nu bepaald worden welke de onderliggende woordsequentie is die deze vectoren heeft gegenereerd. Het woord $\hat{W}$ met de grootste kans om deze observatievectoren te genereren wordt gekozen:

$$\hat{W} = \arg\max_{W} P(W|\mathbf{Y_1}, ..., \mathbf{Y_k}) \tag{2.1}$$

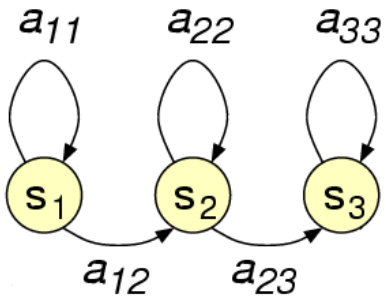
met $\mathbf{Y_1}, ..., \mathbf{Y_k}$ de k observatievectoren en W een bepaald woord. Deze formule kan volgens de regel van Bayes herschreven worden:

$$\hat{W} = \arg\max_{W} \frac{P(\mathbf{Y_1}, ..., \mathbf{Y_k}|W)P(W)}{P(\mathbf{Y_1}, ..., \mathbf{Y_k})} \tag{2.2}$$

Hierbij is de noemer onafhankelijk van de keuze van $\hat{W}$. Deze kan dus weggelaten worden. $P(W)$ is de kans op het voorkomen van een bepaald woord en wordt bepaald aan de hand van taalmodellen. De term $P(\mathbf{Y_1}, ..., \mathbf{Y_k}|W)$ geeft de kans dat een bepaald woord een bepaalde sequentie van observatievectoren veroorzaakt. Deze kunnen berekend worden indien er voor elk woord een model opgesteld wordt dat vectorsequenties genereert. Het woordmodel met de grootste waarschijnlijkheid om deze vectoren te genereren bepaalt het woord dat wordt gekozen. De woordmodellen worden beschreven aan de hand van eerste-orde verborgen Markov-ketens ('*Hidden Markov Models*' of HMM). Ook fonemen worden via HMM's gemodelleerd.

[2]De spraakherkenner die werd gebruikt in de experimenten van dit eindwerk (de ESAT-speech recogniser) gebruikt geen DCT als laatste stap. In de plaats hiervan worden twee complexe algoritmes gebruikt, die zorgen voor een optimale scheidbaarheid en een optimale selectie van de coëfficiënten. Voor de decorrelatie wordt 'least squares decorrelation' gebruikt [3]. Voor het selecteren van de beste kenmerken uit de kenmerkenvector wordt 'linear discriminant analysis' (LDA) gebruikt [4, 5]. Dit gebeurt na het toevoegen van de eerste en tweede afgeleide aan de kenmerkenvector (cfr. de laatste stap van de voorverwerking).

Een Markov-keten bestaat uit een aantal toestanden waartussen zich een aantal overgangen bevinden. Elke overgang kan met een kans a_{ij} optreden, waarbij i het label is van de toestand waarin de pijl vertrekt, en j die van de toestand waar de pijl toekomt (zie figuur 2.2). Deze kans is enkel afhankelijk van de toestand waarin men zich op dat moment bevindt en dus onafhankelijk van hoe men in deze toestand is gekomen. Er wordt gestart in een bepaalde toestand en op elke kloktik wordt een overgang gemaakt naar een volgende toestand. Met elke toestand wordt bovendien een bepaalde kansdichtheidsfunctie van dimensie N geassocieerd, waarbij N de dimensie is van de kenmerkenvectoren van de spraakherkenner. Op basis van deze kansdichtheidsfunctie wordt op elke kloktik een vector met dimensie N gegenereerd. De keuze van deze vector is dus enkel afhankelijk van de toestand waarin men zich op dat moment bevindt. Op deze manier wordt door het model een vectorsequentie gegenereerd. Indien enkel deze vectorsequentie geobserveerd wordt, kan niet bepaald worden wat de sequentie van toestanden was die werd doorlopen, aangezien elke toestand elke vector kan genereren. Aangezien de toestanden dus verborgen zijn wordt dit een 'verborgen' Markov-keten genoemd.

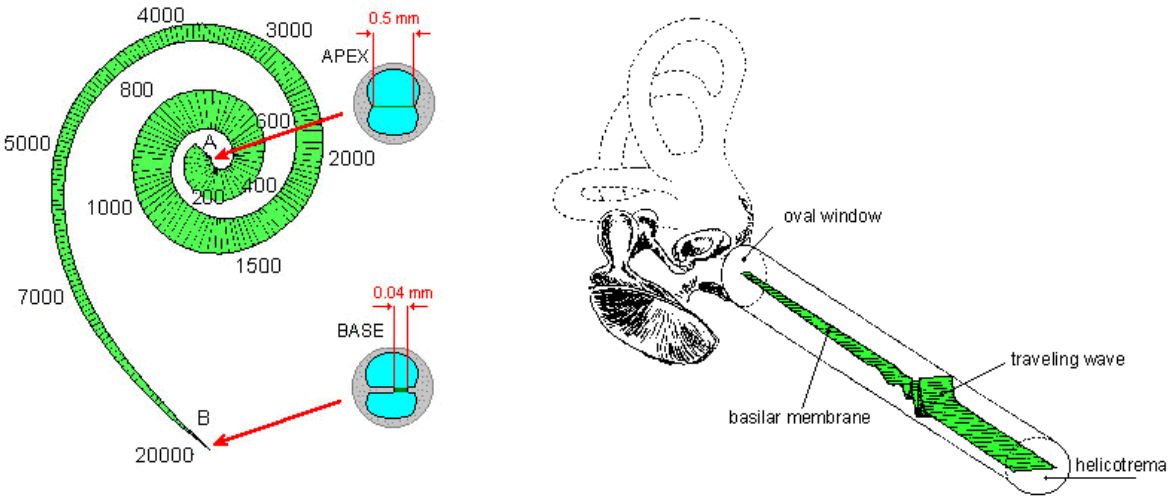


Figuur 2.2: Een voorbeeld van een eerste orde Markov model

Als een bepaalde geobserveerde vectorsequentie gegeven is, kan de kans berekend worden dat een bepaalde HMM deze vectorsequentie heeft veroorzaakt. Dit kan aan de hand van de transitiekansen en de kansdichtheidsfuncties in elke toestand. Het komt er nu op aan om het HMM te vinden waarbij deze kans maximaal is. Dit is een heel rekenintensieve taak en gebeurt aan de hand van snelle zoekalgoritmes.

Het opstellen van de verschillende foneem- en woordmodellen (het trainen van de herkenner) is vaak een moeilijk probleem wegens de schaarste aan data. Voor het modelleren van fonemen worden typisch drie toestanden gebruikt. De kansen voor de verschillende transities en voor de selectie van de initiële toestand moeten bepaald worden uit de trainingsdata. Dit is ook het geval voor de kansdichtheidsfuncties in elke toestand voor het genereren van de vectoren. Deze worden gemodelleerd als een gewogen som van een aantal N-dimensionale Gaussianen. De Gaussianen worden beschreven met diagonale covariantiematrices om het aantal te schatten parameters te beperken. De technieken om al deze parameters te bepalen vallen buiten het bestek van deze thesis.

Merk op dat het schatten van de parameters van de HMM's een vorm is van 'supervised learning'. Er wordt namelijk gebruik gemaakt van een manuele classificatie van fonemen. Een HMM voor foneem [e:] krijgt op deze manier enkel data die afkomstig is van uitingen van dit foneem. In het tweede experiment binnen dit eindwerk wordt een poging ondernomen om tot een automatische classificatie van fonemen te komen zonder a-priorische kennis (zie hoofdstuk 5).



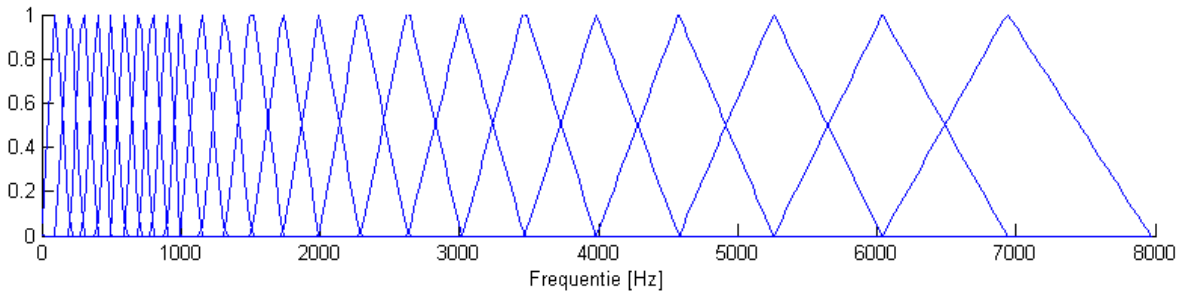
Figuur 2.3: Een schematische voorstelling van het basilair membraan (links). De waarden 200 tot 20000 zijn uitgedrukt in Hz, en geven aan op welke frequentie het membraan op die plaats het sterkst meetrilt. Op de rechter figuur wordt het ontrolde basilair membraan weergegeven. Het membraan wordt hier geëxciteerd door een golf van ongeveer 1000Hz.

2.3 De MEL-filterbank

Veel spraakherkenners gebruiken de MEL-filterbank voor kenmerken-extractie. Deze filterbank is gebaseerd op het menselijk auditief systeem. Het modelleert de frequentiegevoeligheid van het menselijk oor. Het oor voert namelijk een niet-uniforme frequentie-analyse uit. De vibraties van het trommelvlies worden op een mechanische manier overgebracht op het basilair membraan. Dit is een langwerpig membraan dat opgerold is en zich in het slakkenhuis bevindt. In figuur 2.3 wordt dit membraan schematisch weergegeven[3]. Het basilair membraan is samengesteld uit vezels die onder spanning staan. Indien deze vezels op de juiste frequentie geëxciteerd worden, zullen ze meetrillen met de invallende golf. Het membraan verandert in breedte en in dikte langsheen zijn lengte. In het brede deel is de densiteit van vezels veel kleiner dan in het smalle gedeelte aan het einde van het membraan. Dit betekent dat het brede gedeelte veel soepeler is en dus vooral zal meetrillen met de laag-frequente golven. Het smalle gedeelte daarentegen zal vooral met hoog-frequente golven meetrillen. Het trillend membraan exciteert op zijn beurt de haarcellen die zich langsheen de lengte van het membraan bevinden. Deze cellen zetten de mechanische signalen om naar chemische potentiaal-signalen die vervolgens aan de hersenen worden doorgegeven.

Net zoals de Fourier-transformatie voert het basilair membraan dus een frequentie-analyse uit. Toch is de frequentie-analyse van het basilair membraan te complex om met een Fourier-transformatie gemodelleerd te worden. Een belangrijk verschil is de niet-uniforme frequentieresolutie van het basilair membraan. De mens heeft namelijk een betere frequentieresolutie op lage frequenties dan op hogere frequenties. Dit kan men aantonen aan de hand van luistertesten: als men de frequentie van een welbepaalde toon lichtjes laat variëren, dan moet de variatie in frequentie bij hoge frequenties groot zijn alvorens men een verschil kan waarnemen. Hoe lager de frequentie, hoe kleiner de nodige frequentievariatie.

[3]Figuur overgenomen uit http://www.vimm.it/cochlea/cochleapages/theory/. Deze site bevat een gedetailleerde uitleg over de werking van het oor.



Figuur 2.4: De MEL-filterbank op basis van de Davis & Mermelstein benadering van de MEL-schaal

De niet-uniforme frequentieresolutie van het basilair membraan is net wat de MEL-filterbank probeert te modelleren. Op basis van gedetailleerd onderzoek werd de MEL-schaal ingevoerd die de frequentieresolutie van het menselijk gehoorsysteem benadert. Dit is een niet-lineaire mapping van de frequentie-as en wordt gedefinieerd als:

$$\text{MEL}(f) = 2595 \log (1 + \frac{f}{700}) \tag{2.3}$$

waarbij f de frequentie is in Hz. Deze schaal wordt vaak benaderd door een lineair verloop tot 1000 Hz en een logaritmisch verloop boven 1000 Hz (Davis & Mermelstein benadering).

In figuur 2.4 wordt de MEL-filterbank weergegeven op een lineaire frequentie-as. Deze bestaat uit een aantal driehoekige vensters met een breedte van ongeveer 200 MEL. De vensters overlappen en volgen elkaar op met ongeveer 1 venster per 100 MEL[4]. Merk op dat de filters breder worden bij hogere frequenties. Dit modelleert het feit dat de frequentieresolutie van het oor in deze gebieden veel kleiner is. De gesommeerde gewogen energie binnen een subband van de MEL-filterbank vormt een MEL-coëfficiënt. Deze coëfficiënten worden vaak gebruikt als kenmerken voor het beschrijven van menselijke spraak. Het blijkt dat deze kenmerken goede herkenningsresultaten opleveren. De MEL-coëfficiënten blijken een goed evenwicht te vormen tussen rekencomplexiteit en robuustheid.

2.4 Besluit

In dit hoofdstuk werd een korte inleiding gegeven omtrent de werking van een spraakherkenningssysteem. Deze verschaft een basiskennis voor de lezer die niet vertrouwd is met de materie. In het vervolg van deze tekst zullen concepten aangehaald worden die in dit hoofdstuk werden beschreven.

Vooral de MEL-filterbank is een belangrijk concept waarnaar in deze tekst nog vaak zal verwezen worden. De MEL-filterbank wordt in spraakherkenners vaak gebruikt als kenmerken-extractor. Het is belangrijk op te merken dat het onderzoek in het domein van de spraakherkenning al een lange geschiedenis achter de rug heeft en dat deze filterbank als een van de beste alternatieven naar voor is getreden voor kenmerken-extractie. De MEL-filterbank vindt een goed evenwicht tussen robuustheid en complexiteit. Hij is gebaseerd op de werking van het menselijk gehoorsysteem. In hoofdstuk 4 wordt gezocht naar een beter alternatief voor deze filterbank. Dit zal gebeuren aan de hand van niet-negatieve matrix-factorisatie algoritmes.

[4]In feite toont figuur 2.4 niet de echte MEL-filterbank, aangezien een benadering van de MEL-schaal werd gebruikt (de Davis & Mermelstein benadering).

Hoofdstuk 3

Matrix-factorisatietechnieken

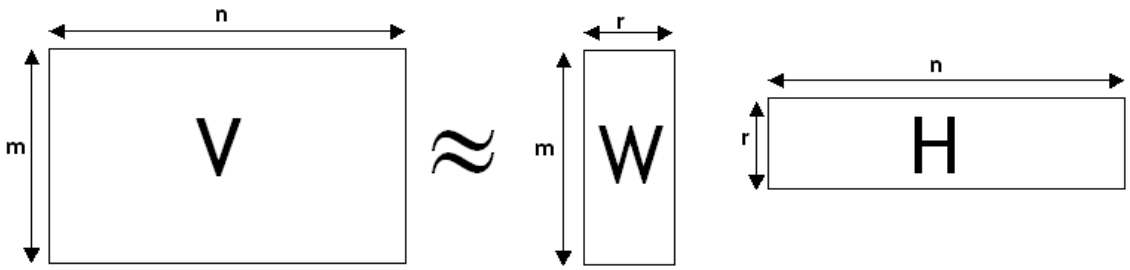
In dit hoofdstuk worden alle matrix-factorisatietechnieken besproken die in de experimenten voor dit eindwerk werden gebruikt. De bedoeling van deze technieken is het benaderen van een matrix $\mathbf{V}$ door een product van twee matrices van lagere dimensie. Formeel betekent dit:

$$\mathbf{V} \approx \mathbf{W}.\mathbf{H} \tag{3.1}$$

met $\mathbf{V}$ een $(m \times n)$ matrix, $\mathbf{W}$ een $(m \times r)$ matrix en $\mathbf{H}$ een $(r \times n)$ matrix waarbij $r \leq m$. Dit is schematisch weergegeven in figuur 3.1. Het spreekt voor zich dat een dergelijke factorisatie de matrix $\mathbf{V}$ slechts nauwkeurig kan reconstrueren indien de data in $\mathbf{V}$ een zekere structuur heeft. Deze structuur is latent aanwezig in de data, maar de achterliggende processen die de elementen in de matrix genereren zijn niet rechtstreeks observeerbaar. Via factorisatietechnieken wordt gepoogd om deze latente structuur bloot te leggen. Aangezien het om matrixvermenigvuldigingen gaat wordt verondersteld dat de latente structuur lineair is. De kolommen van $\mathbf{H}$ bevatten de gewichten waarmee de basisvectoren in de kolommen van $\mathbf{W}$ lineair worden samengesteld om de kolommen van $\mathbf{V}$ te reconstrueren. Dit komt neer op een dimensiereductie van een m-dimensionale ruimte naar een r-dimensionale ruimte.

Bepaalde technieken zullen er ook voor zorgen dat matrices $\mathbf{W}$ en $\mathbf{H}$ enkel niet-negatieve elementen bevatten (in de veronderstelling dat $\mathbf{V}$ geen negatieve elementen bevat). In dat geval wordt gesproken over niet-negatieve matrix-factorisatie. Niet-negativiteitsvoorwaarden zijn interessant omdat dergelijke factorisaties een deel-gebaseerde structuur blootleggen. De kolommen van matrix $\mathbf{W}$ kunnen dan namelijk geïnterpreteerd worden als bouwblokken die alleen op een additieve manier mogen samengesteld worden om de kolommen van $\mathbf{V}$ te reconstrueren. Vaak leidt dit tot basisvectoren met een spaarse structuur. In [6] wordt geargumenteerd dat de patronen die via een niet-negatieve matrix-factorisatie worden gevonden beter overeenkomen met menselijke perceptie dan de patronen die door matrix-factorisatie technieken zonder niet-negativiteitsvoorwaarden worden gevonden. Aangezien de menselijke perceptie van spraak het hoofdonderwerp is binnen dit eindwerk, zullen vooral niet-negatieve factorisatietechnieken gebruikt worden.

In sectie 3.1 wordt een vaak gebruikte matrix-factorisatietechniek geïntroduceerd: de singuliere-waardenontbinding (SWO). De voor- en nadelen van deze techniek zullen uitgelegd worden aan de hand van een concrete toepassing ervan: latent semantische analyse (LSA). Dit is een techniek die gebruikt wordt om artikels of teksten te classificeren in een aantal latente klassen. LSA wint sterk aan belang binnen het domein van de spraakherkenning, aangezien deze techniek op een zelflerende manier semantische relaties tussen woorden en zinnen kan ontdekken. Dit is interessant voor het opstellen van



Figuur 3.1: Schematische voorstelling van een matrix-factorisatie

taalmodellen.

Hoewel LSA op het eerste zicht weinig te maken heeft met de specifieke experimenten van dit eindwerk, is een bespreking van LSA-technieken toch nuttig. De PLSA-factorisatie die in sectie 3.2 wordt afgeleid is namelijk ontstaan binnen het LSA-domein, en is het makkelijkst te interpreteren via deze concrete toepassing. Bovendien is er een duidelijke analogie tussen PLSA en het tweede deelprobleem in dit eindwerk (zie hoofdstuk 5). De opstelling en de doelstelling van de experimenten in hoofdstuk 5 is dan ook intuïtief makkelijker te begrijpen indien de lezer vertrouwd is met de theorie van PLSA en LSA in het algemeen.

In sectie 3.3 wordt een derde techniek geïntroduceerd: niet-negatieve matrix-factorisatie (NMF). Er zal blijken dat PLSA en NMF twee equivalente factorisatietechnieken zijn, ondanks hun verschillende theoretische achtergrond.

3.1 Singuliere-waardenontbinding

3.1.1 Wiskundige achtergrond

De singuliere waardenontbinding (SWO) is een orthogonale ontbinding van de $(m \times n)$ matrix $\mathbf{V}$ in de vorm

$$\mathbf{V} = \mathbf{U}.\mathbf{\Sigma}.\mathbf{H} \tag{3.2}$$

waarbij $\mathbf{\Sigma}$ een $(n \times n)$ of een $(m \times m)$ matrix is met op de diagonaal de singuliere waarden van de matrix $\mathbf{V}$. De singuliere waarden zijn gelijk aan de vierkantswortel van de eigenwaarden van de matrix $\mathbf{V}^T\mathbf{V}$ of $\mathbf{V}\mathbf{V}^T$ (beiden hebben dezelfde niet-nul eigenwaarden). De matrices $\mathbf{U}$ en $\mathbf{H}$ zijn orthonormaal en bevatten de overeenkomstige eigenvectoren van $\mathbf{V}^T\mathbf{V}$ en $\mathbf{V}\mathbf{V}^T$. Voor een gedetailleerde wiskundige beschrijving wordt verwezen naar de literatuur.

Er kan bewezen worden dat SWO orthonormale richtingen zoekt met maximale variantie in de n-dimensionale ruimte met datapunten (kolommen van $\mathbf{V}$). De singuliere waarden in matrix $\mathbf{\Sigma}$ zijn gelijk aan de standaarddeviaties van de datapunten in de richting van de overeenkomstige eigenvector. Enkel de r grootste singuliere waarden worden behouden en de andere elementen van $\mathbf{\Sigma}$, met hun overeenkomstige rijen en kolommen in $\mathbf{U}$ en $\mathbf{H}$, worden verwijderd. Aangezien de datapunten een kleine variantie hebben in de richtingen die verwijderd worden, gaat er weinig informatie verloren. Er kan bewezen worden dat het matrixproduct nu een beste rang-r benadering van de matrix $\mathbf{V}$ genereert. Dit betekent dat dit de matrix van rang r oplevert met een minimale kwadratische afstand tot de

oorspronkelijke matrix $\mathbf{V}$. SWO vindt dus steeds het globale optimum voor het 'Mean-Squared-Error' criterium (MSE)[1]. Dit is meteen het grootste voordeel van deze factorisatietechniek.

3.1.2 Latent Semantische Analyse (LSA)

Ter illustratie wordt kort een techniek besproken die van SWO gebruik maakt: de Latent Semantische Analyse (LSA) [7]. Dit is een techniek die artikels automatisch classificeert in een aantal latente klassen, zonder dat deze klassen op voorhand gedefinieerd zijn. Zo zullen artikels die over gelijkaardige onderwerpen gaan automatisch in eenzelfde latente klasse terechtkomen. Dit kan gebeuren door een *co-occurrence* matrix $\mathbf{V}$ te factoriseren. Deze matrix bevat *'word-counts'* voor elk artikel. Formeel betekent dit:

$$\mathbf{V}_{ij} = n(d_i, w_j) \tag{3.3}$$

met $n(d_i, w_j)$ het aantal keer dat woord w_j voorkomt in document d_i. Indien m artikels moeten geclassificeerd worden op basis van n sleutelwoorden, dan is $\mathbf{V}$ een $(m \times n)$ matrix. Deze matrix kan nu gefactoriseerd worden via SWO. De r grootste singuliere waarden worden behouden. De keuze van r bepaalt het aantal latente klassen waarin de artikels worden geclassificeerd. De kolommen van $\mathbf{U}$ en de rijen van $\mathbf{H}$ kunnen nu geassocieerd worden met deze latente klassen. De waarden in de rijen van matrix $\mathbf{H}$ geven aan in hoeverre het corresponderende woord de respectievelijke latente klasse verklaart. Het woord 'cel' zal bv. een hoge waarde krijgen in de rij van de latente klasse over celbiologie, maar ook een hoge waarde in de rij van de latente klasse over het gevangenisleven. Analoog geven de waarden in de kolommen van matrix $\mathbf{U}$ aan in welke mate het corresponderende artikel in de respectievelijke latente klasse thuishoort. De semantische relaties tussen woorden en tussen verschillende artikels onderling worden op deze manier blootgelegd.

Er is een reële kans dat twee artikels A en B weinig of geen woorden gemeenschappelijk hebben, hoewel ze toch over gelijkaardige onderwerpen gaan. Toch is de kans groot dat de LSA-techniek deze artikels in dezelfde latente klasse onderbrengt. Dit is mogelijk als artikel A gemeenschappelijke woorden heeft met artikel C, dat op zijn beurt gemeenschappelijke woorden heeft met artikel B. Op deze manier ontstaat er dus een link tussen artikels A en B, via een omweg langs artikel C.

Hoewel SWO een goede techniek blijkt te zijn om semantische relaties tussen woorden en teksten te vinden, geeft de theoretische fundering voor het gebruik van SWO voor deze toepassing weinig voldoening. Er is namelijk een gebrek aan een onderliggend model. De matrix met de gewichten die aangeven in hoeverre een bepaalde tekst of woord bijdraagt tot een bepaalde klasse bevat bovendien negatieve gewichten. Het is moeilijk om dit te interpreteren. Wat is bijvoorbeeld de betekenis van het feit dat artikel A met een gewicht -2 tot de klasse X behoort? Een mogelijke interpretatie is dat artikel A helemaal niet tot klasse X behoort. Deze interpretatie is echter niet volledig, want een negatief gewicht betekent dat er in andere klasses vermogen wordt weggehaald.

[1] Een orthonormale basis voldoet aan de truncatie-eigenschap. Dit betekent dat de kwadratische fout op de reconstructie gelijk is aan de kwadratische som van alle gewichten die niet voor de reconstructie worden gebruikt. SWO zoekt orthonormale richtingen met maximale variantie. Indien de richtingen met grootste variantie (grootste singuliere waarden) worden gebruikt voor de reconstructie, impliceert de truncatie-eigenschap dat een globaal minimum wordt gevonden voor de MSE-kostfunctie.

3.2 Probabilistische Latent Semantische Analyse (PLSA) en het EM-algoritme

3.2.1 Doel

Probabilistische Latent Semantische Analyse (PLSA) heeft dezelfde doelstellingen als de standaard LSA-techniek, namelijk het classificeren van teksten in een aantal latente klassen. In tegenstelling tot gewone LSA heeft PLSA een statistisch gefundeerd onderliggend model. Bovendien is PLSA een niet-negatieve matrix-factorisatietechniek, waardoor er geen interpretatieproblemen zijn met negatieve gewichten.

Net zoals standaard-LSA kan PLSA semantische verbanden vinden tussen teksten, zelfs als deze geen woorden gemeenschappelijk hebben. Dit is een belangrijke eigenschap waarop de experimenten voor het tweede deelprobleem binnen dit eindwerk gebaseerd zijn[2].

Hoewel de doelstelling van PLSA gelijkaardig lijkt aan clustering, is er een belangrijk verschil. In clusteringsalgoritmes worden clusters gezocht die een aantal datapunten volledig verklaren. Op deze manier behoort een tekst steeds tot slechts één klasse. PLSA daarentegen is een aspect-model. Dit betekent dat de datapunten (teksten of woorden) kunnen verklaard worden aan de hand van meerdere aspecten. Op deze manier kan een tekst bv. voor 30% tot klasse A en voor 70% tot klasse B behoren. Ook voor spraakdata is dit een wenselijke eigenschap. Wegens co-articulatie zullen klanken vaak tussen twee verschillende fonemen liggen. Clusteringsalgoritmes worden hierdoor in verwarring gebracht, aangezien dit klanken zijn die rond de grens tussen twee foneemklassen liggen. PLSA houdt hier rekening mee.

Clusters en aspecten zijn dus twee verschillende begrippen die op een ander model gebaseerd zijn. De gevonden aspecten zijn dan ook vaak erg verschillend van de clusters die met een clusteringsalgoritme werden bepaald.

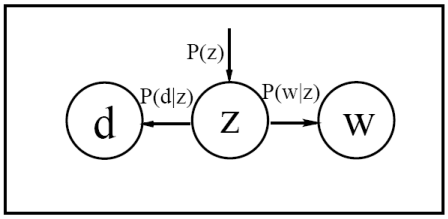
3.2.2 Theoretisch model

PLSA is een iteratieve niet-negatieve factorisatietechniek die in [8] wordt afgeleid aan de hand van een statistisch aspect-model [9] waarvan de parameters met het EM-algoritme worden geschat [10]. Hieronder wordt deze afleiding iets uitgebreider behandeld.

De *word-count* matrix gedefinieerd door (3.3) is een representatie van een reeks observaties (d, w), waarbij (d, w) het voorkomen van woord w in document d aanduidt. In PLSA wordt verondersteld dat elk van deze observaties het gevolg is van een niet-observeerbare variabele z die slechts een beperkt aantal discrete waarden kan aannemen: $z \in Z = \{z_1, \ldots, z_r\}$, waarbij z_i een bepaalde latente klasse of een bepaald aspect voorstelt. Het feit dat woord w_j in document d_i voorkomt is volledig te verklaren door de onderliggende latente variabele die deze observatie heeft gegenereerd.

Er wordt een model opgesteld dat onder deze veronderstelling een aantal observaties genereert. Dit wordt voorgesteld in figuur 3.2. Het model selecteert een latente variabele z met kans $P(z)$. Daarna wordt er een woord w en een document d gegenereerd met een kans $P(w|z)$, respectievelijk $P(d|z)$. Deze kansen zijn dus alleen afhankelijk van de gekozen latente variabele z. Het komt er nu op aan om

[2] Zie hoofdstuk 5: Zelflerend foneem-model



Figuur 3.2: Het aspect-model genereert observaties (d, w) afhankelijk van de gekozen latente variabele z.

de kansen $P(z)$, $P(w|z)$ en $P(d|z)$ te schatten opdat de waarschijnlijkheid dat het model de gegeven observatiematrix $\mathbf{V}$ genereert, maximaal is.

De waarschijnlijkheid dat het model de matrix $\mathbf{V}$ genereert is

$$P(\mathbf{V}) = \prod_{i,j} P(d_i, w_j)^{n(d_i,w_j)} \tag{3.4}$$

met $n(d_i, w_j)$ het aantal keer dat woord w_j in document d_i voorkomt en $P(d_i, w_j)$ de kans dat een observatie (d_i, w_j) wordt gegenereerd. Maximalisatie van (3.4) is equivalent aan het maximaliseren van de log-likelihood:

$$\Lambda(\lambda) = \sum_{i,j} n(d_i, w_j) \log P(d_i, w_j) \tag{3.5}$$

met λ de verzameling van de te schatten parameters die samen de kans $P(d_i, w_j)$ bepalen. Volgens figuur 3.2 wordt dit:

$$\Lambda(\lambda) = \sum_{i,j} n(d_i, w_j) \log \left(\sum_{z \in Z} P(z)\ P(d_i|z)\ P(w_j|z) \right) \tag{3.6}$$

waarbij de kansen $P(z)$, $P(d_i|z)$ en $P(w_j|z)$ de te schatten parameters zijn. Aangezien de log-likelihood moeilijk rechtstreeks gemaximaliseerd kan worden, gebeurt het schatten van deze parameters iteratief aan de hand van het EM-algoritme [10]. Dit algoritme start met een initiële keuze voor λ en probeert deze op een iteratieve manier te verbeteren. In elke iteratiestap wordt de volgende hulpfunctie gemaximaliseerd:

$$H(\lambda, \lambda_0) = \sum_{i,j} n(d_i, w_j) \sum_{z \in Z} P_0(z|d_i, w_j) \log \left(\frac{P(z)\ P(d_i|z)\ P(w_j|z)}{P_0(z|d_i, w_j)} \right) \tag{3.7}$$

Een index 0 duidt aan dat het om de huidige geschatte parameters gaat. De bedoeling is om een λ te bepalen aan de hand van de huidige schatting λ_0 opdat (3.6) groter wordt. Er kan bewezen worden dat $H(\lambda, \lambda_0) \leq \Lambda(\lambda)$, en dat beide functies elkaar raken als $\lambda = \lambda_0$ [10]. Dit betekent dat een maximalisatie van $H(\lambda, \lambda_0)$ de likelihood $\Lambda(\lambda)$ ook doet stijgen (of onveranderd laat). Als de nieuwe schatting gelijk is aan de oude schatting betekent dit dat beide functies elkaar raken in een stationair punt van $H(\lambda, \lambda_0)$. Aangezien beide functies op dit punt dezelfde afgeleide hebben, impliceert dit dat ook een stationair punt van de log-likelihood $\Lambda(\lambda)$ bereikt is.

De hulpfunctie $H(\lambda, \lambda_0)$ kan in tegenstelling tot $\Lambda(\lambda)$ op een analytische manier geoptimaliseerd worden. Dit gebeurt via de methode van de Lagrange-vermenigvuldigers. Op deze manier worden de

volgende update-formules bekomen:

$$P(w_j|z) = \frac{\sum_i n(d_i, w_j)P(z|d_i, w_j)}{\sum_{i,j} n(d_i, w_j)P(z|d_i, w_j)} \tag{3.8}$$

$$P(d_i|z) = \frac{\sum_j n(d_i, w_j)P(z|d_i, w_j)}{\sum_{i,j} n(d_i, w_j)P(z|d_i, w_j)} \tag{3.9}$$

$$P(z) = \frac{\sum_{i,j} n(d_i, w_j)P(z|d_i, w_j)}{\sum_{i,j} n(d_i, w_j)} \tag{3.10}$$

waarbij de a posteriori kans $P(z|d_i, w_j)$ uit de huidige schatting wordt bepaald:

$$P(z|d_i, w_j) = \frac{P_0(z)\ P_0(d_i|z)\ P_0(w_j|z)}{\sum_{z \in Z} P_0(z)\ P_0(d_i|z)\ P_0(w_j|z)} \tag{3.11}$$

3.2.3 Matrix-factorisatie op basis van PLSA

Hoewel in PLSA de matrix $\mathbf{V}$ steeds een matrix is met natuurlijke getallen, is dit uiteraard geen voorwaarde voor het toepassen van deze techniek. Indien de matrix $\mathbf{V}$ genormaliseerd wordt volgens

$$\hat{v}_{ij} = \frac{v_{ij}}{\sum_{i,j} v_{ij}} \tag{3.12}$$

dan kan $\hat{\mathbf{V}}$ beschouwd worden als een 2-dimensionale discrete kansverdeling in de discrete stochastische variabelen d en w. Er geldt dus dat $\hat{v}_{i,j} = P(d_i, w_j)$. Aangezien het PLSA-model een maximale waarschijnlijkheid heeft om de data in matrix $\mathbf{V}$ te genereren, betekent dit dat de kansverdeling $P'(d_i, w_j)$ die door de geschatte parameters van dit model wordt beschreven een optimale benadering is van de kansverdeling $P(d_i, w_j)$. Het PLSA-model dat hierboven werd afgeleid kan dus geïnterpreteerd worden als een matrix-factorisatie:

$$\hat{\mathbf{V}} \approx \begin{bmatrix} P(d_1|z_1) & \cdots & P(d_1|z_r) \\ \vdots & \ddots & \vdots \\ P(d_m|z_1) & \cdots & P(d_m|z_r) \end{bmatrix} \begin{bmatrix} P(z_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P(z_r) \end{bmatrix} \begin{bmatrix} P(w_1|z_1) & \cdots & P(w_n|z_1) \\ \vdots & \ddots & \vdots \\ P(w_1|z_r) & \cdots & P(w_n|z_r) \end{bmatrix}$$

of kortweg:

$$\hat{\mathbf{V}} = \hat{\mathbf{U}}.\hat{\mathbf{\Sigma}}.\hat{\mathbf{H}} \tag{3.13}$$

Hierbij is $\hat{\mathbf{V}}$ een $(m \times n)$ matrix, $\hat{\mathbf{\Sigma}}$ een diagonale $(r \times r)$ matrix, $\hat{\mathbf{U}}$ een $(m \times r)$ matrix en $\hat{\mathbf{H}}$ een $(r \times n)$ matrix. Merk op dat (3.13) eenvoudig in de vorm (3.1) kan geschreven worden. Het is arbitrair of de matrix $\hat{\mathbf{\Sigma}}$ tot $\mathbf{W}$, al dan niet tot $\mathbf{H}$ wordt gerekend.

3.2.4 Eigenschappen van de PLSA-factorisatie

Bemerk de analogie tussen (3.13) en (3.2) waarbij slechts de r grootste singuliere waarden werden behouden. Het grote verschil tussen beide factorisaties is echter de kostfunctie die geminimaliseerd wordt. SWO minimaliseert de gemiddelde gekwadrateerde fout (MSE-criterium), terwijl PLSA de voorspellende kracht van het onderliggende model maximaliseert. Dit komt neer op een minimalisatie

van de kruis-entropie of Kullback-Leibler divergentie tussen de kansverdeling van het PLSA-model en de kansverdeling gedefinieerd door $\hat{\mathbf{V}}$.

PLSA lost de tekortkomingen van SWO op. PLSA heeft in tegenstelling tot SWO een onderliggend theoretisch model. Bovendien is de interpretatie van de factorisatie veel intuïtiever. De elementen in de matrices kunnen namelijk als kansen beschouwd worden. Dit betekent dat er geen negatieve gewichten zijn, wat de interpretatie van de blootgelegde structuur explicieter maakt. Via de regel van Bayes kunnen uit $\hat{\mathbf{U}}$ en $\hat{\mathbf{H}}$ makkelijk de a posteriori kansen $P(z|d)$ en $P(z|w)$ berekend worden. Aangezien alle gewichten voor elke variabele sommeren tot 1, kunnen hieruit procentuele bijdrages berekend worden van elk aspect tot elk woord of document. Hoe dichter deze bijdrage bij 1 ligt, hoe sterker het aspect het respectievelijke woord of document verklaart.

Het belangrijkste nadeel van PLSA ten opzichte van SWO is het feit dat PLSA geen globaal optimum garandeert. Indien de likelihood-functie veel lokale maxima heeft, zal PLSA een suboptimale oplossing genereren, afhankelijk van de gekozen (random) initialisatie.

Een ander belangrijk nadeel is het feit dat de waarde voor de parameter r (het aantal latente klassen of aspecten) a priori moet vastgelegd worden. Bij SWO kan de parameter r achteraf bepaald worden en kan men zich voor deze keuze baseren op de waarden van de elementen in de diagonaalmatrix $\mathbf{\Sigma}$. Een overschatting van r heeft tot gevolg dat de gevonden basisvectoren lineair afhankelijk zijn. Dit is problematisch indien de kolommen van $\mathbf{H}$ bv. als kenmerkenvectoren gebruikt worden om kolommen van de matrix $\mathbf{V}$ te beschrijven. Er zijn dan namelijk verschillende mogelijke kenmerkenvectoren om eenzelfde kolom voor te stellen[3]. Een onderschatting van r geeft uiteraard een suboptimale reconstructie van $\mathbf{V}$. Uit experimenten blijkt dat de energie van de weggevallen basisvectoren dan wordt verdeeld over de r overblijvende basisvectoren.

3.2.5 Implementatie en convergentie-eigenschappen

Voor de implementatie wordt (3.13) herschreven in de vorm (3.1) met $\mathbf{W} = \hat{\mathbf{U}}.\hat{\mathbf{\Sigma}}$, $\mathbf{H} = \hat{\mathbf{H}}$ en $\mathbf{V} = \hat{\mathbf{V}}$. De matrices $\mathbf{W}$ en $\mathbf{H}$ worden als positieve random matrices geïnitialiseerd. Aangezien de rijen van $\hat{\mathbf{H}}$ steeds moeten sommeren tot 1, wordt op de rijen van $\mathbf{H}$ de volgende normalisatie toegepast:

$$\mathbf{H}_{ij} = \frac{\mathbf{H}_{ij}}{\sum_j \mathbf{H}_{ij}} \tag{3.14}$$

Als er aan deze normalisatie voldaan is, kan er bewezen worden dat de volgende updateformules equivalent zijn aan de updates van de parameters van het PLSA model [11]:

$$\mathbf{W}_{ij}^{(t+1)} = \mathbf{W}_{ij}^{(t)} \sum_a \frac{\mathbf{H}_{ja}^{(t)} \mathbf{V}_{ia}}{(\mathbf{W}^{(t)}\mathbf{H}^{(t)})_{ia}} \quad , \qquad \mathbf{H}_{ij}^{(t+1)} = \mathbf{H}_{ij}^{(t)} \frac{\sum_a \frac{\mathbf{W}_{ai}^{(t)} \mathbf{V}_{aj}}{(\mathbf{W}^{(t)}\mathbf{H}^{(t)})_{aj}}}{\sum_a \mathbf{W}_{ai}^{(t+1)}} \tag{3.15}$$

waarbij t het aantal uitgevoerde iteraties aangeeft.

Experimenten tonen aan dat PLSA inderdaad in staat is om de onderliggende structuur te ontdekken in artificiële data-matrices van onvolledige rang. Ook als ruis wordt toegevoegd aan de matrix $\mathbf{V}$

[3] Voor reële spraakdata is een overschatting van r niet mogelijk omdat $\mathbf{V}$ normaal gezien van volle rang zal zijn. Toch zorgt een te grote waarde voor r ervoor dat er minder structuur wordt ontdekt in de matrix.

kan PLSA de onderliggende basisvectoren terugvinden waarmee $\mathbf{V}$ werd opgebouwd. Toch blijft het algoritme regelmatig vastzitten in lokale optima[4].

Tot slot worden een aantal eigenschappen geponeerd die op empirische basis werden vastgesteld:

- De rekentijd nodig per iteratie is lineair afhankelijk van zowel r (aantal aspecten), m (aantal rijen van $\mathbf{V}$) als n (aantal kolommen van $\mathbf{V}$). Dit blijkt ook uit een formele complexiteitsanalyse.
- De resultaten zijn sterk afhankelijk van de initialisatie. In veel gevallen wordt een suboptimale oplossing gevonden.
- De convergentie naar een maximale likelihood gebeurt in het begin redelijk snel, tot op het moment dat een oplossing wordt gevonden die redelijk dicht bij de goede oplossing ligt. Vanaf dan convergeert het algoritme heel traag naar de goede oplossing. Soms verschijnt er plots een significante stijging van de likelihood als deze al een tijdje geconvergeerd lijkt te zijn. Dit fenomeen is uitzonderlijk en is bij spraakdata nooit voorgekomen.
- Convergentie van de likelihood-functie impliceert niet dat de getallen in de matrices $\mathbf{W}$ en $\mathbf{H}$ geconvergeerd zijn. Het blijkt dat de elementen in deze matrices nog sterk variëren, zelfs als de likelihood-functie al enige tijd geconvergeerd is.

3.3 Niet-negatieve matrix-factorisatie (NMF)

3.3.1 Optimalisatiecriteria

Het NMF-algoritme zoekt een factorisatie van de vorm (3.1) die een bepaalde kostfunctie minimaliseert onder niet-negativiteitsvoorwaarden. In de literatuur over NMF worden twee verschillende kostfuncties gebruikt, gebaseerd op respectievelijk de gemiddelde kwadratische fout (MSE-criterium) en de kruisentropie (divergentie-criterium).

Stel de reconstructie van de matrix $\mathbf{V}$ voor door matrix $\mathbf{X}$. Het MSE-criterium wordt dan gedefinieerd als

$$\|\mathbf{V} - \mathbf{X}\|^2 = \sum_{i,j} (\mathbf{V}_{ij} - \mathbf{X}_{ij})^2 \tag{3.16}$$

terwijl het divergentie-criterium gedefinieerd is als

$$Div(\mathbf{V}||\mathbf{X}) = \sum_{i,j} \left(\mathbf{V}_{ij} \log \frac{\mathbf{V}_{ij}}{\mathbf{X}_{ij}} - \mathbf{V}_{ij} + \mathbf{X}_{ij} \right) \tag{3.17}$$

Indien $\sum_{i,j} \mathbf{V}_{ij} = 1$ en $\sum_{i,j} \mathbf{X}_{ij} = 1$ kunnen deze matrices als kansdichtheidsfuncties beschouwd worden en reduceert (3.17) zich tot de Kullback-Leibler divergentie. Merk op dat (3.16) en (3.17) nul worden indien $\mathbf{V} = \mathbf{X}$.

[4]Indien 1000 factorisaties worden uitgevoerd op een 10×50 matrix $\mathbf{V}$ van rang 5, wordt slechts in 82% van de gevallen een globaal optimum gevonden.

3.3.2 Update regels

De kostfuncties (3.16) en (3.17) worden door het NMF-algoritme op een iteratieve manier geminimaliseerd. De initialisatie van de matrices $\mathbf{W}$ en $\mathbf{H}$ is willekeurig, maar mag enkel positieve elementen bevatten. Daarna worden een aantal multiplicatieve updates uitgevoerd op deze matrices, totdat de kostfunctie geconvergeerd is.

In [12] wordt bewezen dat de MSE-kostfunctie (3.16) niet-stijgend is onder de volgende multiplicatieve update regels:

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{(\mathbf{W}^T\mathbf{V})_{a\mu}}{(\mathbf{W}^T\mathbf{W}\mathbf{H})_{a\mu}} \quad , \quad \mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{(\mathbf{V}\mathbf{H}^T)_{ia}}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ia}} \tag{3.18}$$

Bovendien kan er bewezen worden dat de matrices $\mathbf{W}$ en $\mathbf{H}$ invariant zijn onder deze update als en slechts als deze in een stationair punt liggen van de MSE-kostfunctie. Merk op dat er nooit negatieve elementen gegenereerd worden in de matrices $\mathbf{W}$ en $\mathbf{H}$ indien matrix $\mathbf{V}$ geen negatieve elementen bevat.

In [12] wordt bovendien bewezen dat identieke eigenschappen gelden voor het divergentiecriterium onder de volgende update regels:

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{\sum_i (\mathbf{W}_{ia}\mathbf{V}_{i\mu})/(\mathbf{W}\mathbf{H})_{i\mu}}{\sum_k \mathbf{W}_{ka}} \quad , \quad \mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{\sum_\mu \mathbf{H}_{a\mu}\mathbf{V}_{i\mu}/(\mathbf{W}\mathbf{H})_{i\mu}}{\sum_v \mathbf{H}_{av}} \tag{3.19}$$

De multiplicatieve updates (3.18) en (3.19) zijn equivalent aan een additieve update volgens de methode van de steilste helling, waarbij de stapgrootte op een analytische manier afhankelijk is van de huidige schatting van $\mathbf{W}$ en $\mathbf{H}$ [12]. De stapgrootte wordt telkens zo gekozen dat de kostfunctie niet stijgt. Dit vermijdt de iteratieve procedure om een stapgrootte te bepalen die niet leidt tot een stijging van de kost.

Het NMF-algoritme zal afwisselend een update uitvoeren op $\mathbf{W}$ en $\mathbf{H}$. De kostfuncties (3.16) en (3.17) zijn niet-convex in $\mathbf{W}$ en $\mathbf{H}$, waardoor geen globaal optimum gegarandeerd kan worden. Bovendien zijn er oneindig veel mogelijke optimale oplossingen. Indien de matrix $\mathbf{W}$ met een matrix $\mathbf{P}$ achterwaarts wordt vermenigvuldigd, terwijl $\mathbf{H}$ met $\mathbf{P}^{-1}$ wordt vermenigvuldigd, blijft de reconstructie, en bijgevolg ook de kostfunctie, identiek. Om aan de niet-negativiteitsvoorwaarden te blijven voldoen moet $\mathbf{P}$ uiteraard aan bepaalde voorwaarden voldoen.

De doelfuncties (3.16) en (3.17) zijn echter wel convex in $\mathbf{W}$ en $\mathbf{H}$ afzonderlijk. Dus indien één van beide matrices gegeven is, en de updates enkel worden uitgevoerd op de variabele matrix, dan zal steeds een globaal optimum gevonden worden.

3.3.3 Het verband tussen NMF en PLSA

In [11] wordt aangetoond dat een punt invariant is onder de update-regels (3.19) van het NMF-algoritme volgens het divergentie-criterium als en slechts als dit punt invariant is onder de update regels (3.15) van de PLSA-factorisatie. Dit betekent dat een oplossing van NMF met divergentie-criterium ook steeds een oplossing is van PLSA en omgekeerd. Dit is niet verwonderlijk aangezien beide algoritmes een divergentie-criterium minimaliseren.

Er wordt echter niet gewezen op de equivalentie van beide algoritmes. In essentie is het NMF-algoritme met divergentie-criterium quasi identiek aan het PLSA-algoritme. Er zijn slechts drie kleine verschillen:

- De updateregels (3.15) leggen op dat de update van matrix $\mathbf{W}$ steeds voor de update van matrix $\mathbf{H}$ plaatsvindt, aangezien de update van $\mathbf{H}$ zowel van $\mathbf{W}^{(t)}$ als van $\mathbf{W}^{(t+1)}$ gebruik maakt. Voor NMF is de volgorde arbitrair.
- De regels (3.15) maken zowel voor het berekenen van $\mathbf{W}^{t+1}$ als voor $\mathbf{H}^{t+1}$ gebruik van de vorige schatting $\mathbf{W}^{(t)}$ en $\mathbf{H}^{(t)}$. In het NMF-algoritme is de situatie anders. Indien eerst de update van $\mathbf{W}$ wordt uitgevoerd, zal de update van $\mathbf{H}$ op basis van de nieuwe schatting van $\mathbf{W}$ en de oude schatting van $\mathbf{H}$ gebeuren.
- De update van $\mathbf{W}$ in (3.15) bevat slechts één deling. Dit is dankzij het feit dat er in PLSA met kansen wordt gerekend. Dankzij een voorafgaande normalisatie van de rijen van de initialisatie-matrix $\mathbf{H}$, reduceert de tweede deling in de update van $\mathbf{W}$ zich tot een deling door 1. De update van $\mathbf{H}$ is zodanig dat de normalisatie van de rijen in $\mathbf{H}$ steeds behouden blijft.

Ondanks deze drie kleine verschillen blijkt uit experimenten dat beide algoritmes dezelfde convergentie-eigenschappen hebben en steeds dezelfde oplossing vinden bij een identieke initialisatie. De updates volgens het PLSA-algoritme zijn te verkiezen boven het NMF-algoritme omwille van het uitsparen van een normalisatie op $\mathbf{W}$. Indien in het vervolg van deze tekst vermeld wordt dat een factorisatie met NMF volgens het divergentie-criterium werd uitgevoerd, werd de eigenlijke factorisatie met de PLSA update-formules (3.15) berekend.

3.4 Intelligente initialisatie

De convergentiesnelheid van het NMF- of het PLSA-algoritme is sterk afhankelijk van de initialisatie. De matrix $\mathbf{W}$ wordt steeds geïnitialiseerd als een random matrix $\mathbf{W}_0$. Hoewel de initialisatie van matrix $\mathbf{H}$ ook random mag zijn, wordt de volgende initialisatie voor $\mathbf{H}$ toegepast in alle experimenten:

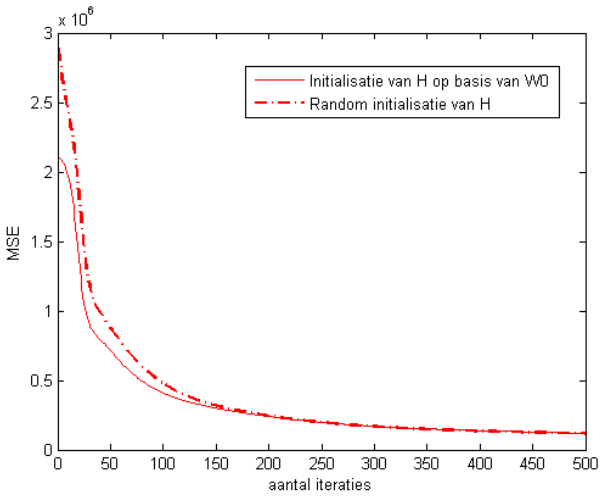
$$\mathbf{H}_0 = \mathbf{W}_0^T.\mathbf{V} \tag{3.20}$$

Dit zorgt ervoor dat de gewichten in de matrix $\mathbf{H}_0$ al in zekere mate aangepast zijn aan de bijbehorende basisvectoren in de kolommen van $\mathbf{W}_0$. Concreet betekent dit dat het gewicht van basisvector $\mathbf{w}_i$ geïnitialiseerd wordt als het product van de L2-norm van de projectie van $\mathbf{w}_i$ op de te reconstrueren kolom van matrix $\mathbf{V}$, met de L2-norm van deze kolom. Indien dit product groot is, geeft dit aan dat er een sterke overeenkomst is tussen de basisvector en de te reconstrueren kolom, aangezien de twee vectoren ongeveer in elkaars verlengde liggen. Bijgevolg zal deze basisvector sterk bijdragen tot de reconstructie van de respectievelijke kolom van $\mathbf{V}$ en wordt er een groot gewicht toegekend in de initialisatie-matrix $\mathbf{H}_0$.

Deze initialisatie zorgt in het algemeen voor een snellere convergentie van de kostfunctie. Indien echter een groot aantal iteraties wordt uitgevoerd, verdwijnt het voordeel van deze initialisatie ten opzichte van een willekeurige initialisatie van $\mathbf{H}$. In beide gevallen wordt dan een gelijkwaardige oplossing gevonden. Dit wordt geïllustreerd in figuur 3.3.

3.5 Besluit

In dit hoofdstuk werden drie matrix factorisatiemethodes besproken. Voor elke techniek werden de belangrijkste eigenschappen aangegeven. SWO heeft als voordeel dat er altijd een globaal optimum



Figuur 3.3: MSE-kostfunctie in functie van het aantal uitgevoerde iteraties bij willekeurige en intelligente initialisatie. De stippellijn geldt voor willekeurige initialisatie van $\mathbf{H}_0$, de volle lijn geldt voor de initialisatie (3.20).

wordt gevonden volgens het MSE-criterium. Hetzelfde criterium kan geoptimaliseerd worden met NMF, maar zonder garantie op het bereiken van een globaal minimum. In tegenstelling tot SWO berekent NMF echter een niet-negatieve factorisatie, waardoor een deel-gebaseerde structuur kan gevonden worden in de data. De matrix $\mathbf{V}$ wordt dan gereconstrueerd door een additieve lineaire combinatie van basisvectoren. Volgens [6] gebeurt menselijke perceptie vaak ook via deel-gebaseerde patronen.

Een factorisatie volgens PLSA is ook steeds niet-negatief en is gebaseerd op een statistisch model. Deze factorisatie biedt het voordeel dat de elementen in de matrices als kansen kunnen beschouwd worden. Er werd gewezen op de gelijkenissen tussen het PLSA-algoritme en het NMF-algoritme volgens het divergentie-criterium. Beide algoritmes zijn quasi identiek, hebben dezelfde convergentie-eigenschappen, en vinden steeds dezelfde oplossingen bij eenzelfde initialisatie.

Hoofdstuk 4

Kenmerken-extractie

4.1 Inleiding

De eerste stap van elke spraakherkenner is het analyseren van het spraaksignaal om er een compacte set kenmerken (*'features'*) uit te extraheren. De kenmerken moeten representatief zijn voor het signaal en moeten dus de informatie bevatten die essentieel is voor het herkennen van het spraaksignaal. Het spreekt voor zich dat de keuze van dergelijke kenmerken enorm bepalend is voor de performantie van de herkenner. State-of-the-art spraakherkenningssystemen gebruiken overwegend MEL-frequentie cepstrale coëfficiënten (MFCC) als kenmerkenset[1]. De overeenkomstige MEL-filterbank is gebaseerd op de frequentieresolutie van het menselijk gehoorsysteem (zie sectie 2.3).

De eerste doelstelling van dit eindwerk is het vinden van een nieuwe kenmerkenset om in spraakherkenning te gebruiken als alternatief voor de MEL-filterbank. In tegenstelling tot de MEL-coëfficiënten die gebaseerd zijn op het menselijk gehoorsysteem, wordt nu geprobeerd om op basis van de spraaksignalen zelf een kenmerkenset te vinden. Hierbij wordt vertrokken vanuit de hypothese dat spraakdata een latente structuur heeft die geëxploiteerd kan worden om tot een compacte voorstelling te komen.

Het is de bedoeling dat matrix-factorisatie algoritmes automatisch goede kenmerken opsporen in een grote hoeveelheid continue spraak. Het opsporen van deze kenmerken gebeurt onder een belangrijke nevenvoorwaarde: er mag buiten de spraaksignalen geen andere informatie gebruikt worden. Er moet een latente structuur gevonden worden in de spraakdata, zonder enige voorkennis over de fonemen of woorden die werden uitgesproken.

Om deze latente structuur bloot te leggen wordt de frequentie-inhoud van heel veel korte stukjes spraak ontbonden via matrix-factorisatie algoritmes. Op deze manier wordt een compacte set van basisvectoren gevonden die een ruimte opspannen waarin het spectrum van het spraakfragment kan beschreven worden. De gebruikte matrix-factorisatietechnieken moeten voor dit experiment in principe niet aan de voorwaarde voldoen van niet-negativiteit. Toch zal blijken dat niet-negatieve matrix-factorisatietechnieken een interessante latente structuur kunnen blootleggen die door SWO niet gevonden wordt.

[1] 'Cepstrale' coëfficiënten slaat hier op het toepassen van de logaritme, gevolgd door een DCT in de voorverwerking van een spraakherkenningssysteem (zie sectie 2.2.1). Dit komt neer op het berekenen van het 'cepstrum' van de MEL-frequentie coëfficiënten.

4.2 Opstelling van het experiment

In de experimenten in dit hoofdstuk wordt een grote matrix $\mathbf{V}$ gefactoriseerd door middel van de factorisatietechnieken die in het vorige hoofdstuk werden geïntroduceerd. De matrix $\mathbf{V}$ bevat meer dan een miljoen kolommen, waarbij elke kolom het frequentiespectrum van een kort spraakfragment bevat. Voor het opstellen van de matrix $\mathbf{V}$ werd de TIMIT databank gebruikt [13, 14]. Deze bevat een grote collectie met continue spraak van verschillende sprekers in de Engelse taal (Engels-Amerikaanse dialecten).

4.2.1 Voorverwerking

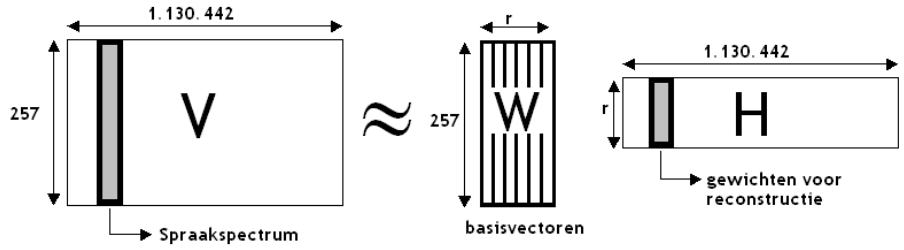
Het spraaksignaal is bemonsterd aan een frequentie van 16000 Hz. Eerst wordt een pre-emphasis toegepast opdat de factorisatie niet gedomineerd zou worden door de lage frequenties (zie sectie 2.2.1). Daarna wordt het signaal opgedeeld in overlappende frames met een framelengte van 25 ms en een frameshift van 10 ms. Op elk frame wordt een Hamming venster en een zero-padding tot 512 punten[2] toegepast, waarna een FFT wordt berekend. Door het nemen van de absolute waarde wordt het frequentiespectrum van elk frame bekomen. Wegens symmetrie is het voldoende om slechts de helft van de punten in het spectrum bij te houden. Het uiteindelijk spectrum bevat dan 257 punten, waarbij het laatste punt overeenkomt met een frequentie van 8000 Hz en het eerste punt met de DC-bijdrage. Afhankelijk van het experiment wordt de absolute waarde van het spectrum gekwadrateerd om het vermogenspectrum van het spraakframe te verkrijgen.

De bekomen spectra bevatten pitch-harmonischen afkomstig van de stembandtrillingen. Aangezien de plaats van deze harmonischen in het spectrum variabel is, kunnen deze niet gemodelleerd worden in een laag-dimensionale ruimte en zullen ze bijgevolg uitgemiddeld worden door de factorisatie. Om deze pitch-harmonischen te verwijderen kan optioneel nog een van de volgende voorverwerkingsoperaties uitgevoerd worden:

1. $Env(\mathbf{X_i})$:
   Deze operator berekent de omhullende van het amplitude- of vermogenspectrum $\mathbf{X_i}$ door de pieken in het spectrum te verbinden met exponentieel dalende curves. Het nadeel van deze methode is dat de energie in het spectrum niet bewaard blijft.
2. $Smooth(\mathbf{X_i})$:
   Deze operator past een spectrale smoothing toe op het vermogenspectrum $\mathbf{X_i}$ door middel van cepstrale coëfficiënten. Dit is een vaak gebruikte techniek in spraakverwerking. Eerst wordt het cepstrum van het vermogenspectrum berekend. Op dit cepstrum wordt een venster toegepast dat de hogere cepstrale coëfficiënten verwijdert. Na een inverse transformatie wordt het oorspronkelijke spectrum bekomen, maar zonder pitch-harmonischen. Het verwijderen van de hoge cepstra is in essentie al een vorm van dimensiereductie en kan dus een bias veroorzaken op de basisvectoren die door NMF worden gevonden.

Zoals in sectie 2.2.2 werd aangehaald, bestaat het akoestisch model van een spraakherkenningssysteem uit een HMM waarvan de kansdichtheidsfuncties gemodelleerd worden door een lineaire combinatie

[2]Zero-padding is een techniek die gebruikt wordt om een spectrum van m punten uit te breiden tot een spectrum van n punten waarbij $n \geq m$. Dit wordt vaak toegepast om het aantal punten uit te breiden naar een getal dat een macht van 2 is. Het gevolg is een interpolatie van de DFT van het oorspronkelijk signaal.



Figuur 4.1: Schematische voorstelling van de matrix-factorisatie voor kenmerken-extractie. Deze leidt tot een dimensie-reductie van de 257-dimensionale ruimte naar een r-dimensionale ruimte.

van Gaussianen. Indien het dynamisch bereik van de trainingsdata te groot is, is het moeilijk om goed passende Gaussianen te bepalen. Daarom wordt steeds de logaritme genomen van de kenmerken-vectoren (zie ook sectie 2.2.1). Deze operatie reduceert het dynamisch bereik van de punten in de kenmerken-ruimte. De logaritmische compressie wordt toegepast op de matrix $\mathbf{H}$ na de factorisatie van $\mathbf{V}$. Voor factorisaties met een MSE-criterium wordt deze compressie toegepast op de matrix $\mathbf{V}$ vóór de factorisatie. Dit is nodig om het optimalisatieprobleem beter te conditioneren (meer hierover in sectie 4.4.1).

De kostfuncties, die door NMF of SWO geminimaliseerd worden, zijn afhankelijk van de energie van de te reconstrueren data[3]. Om te vermijden dat de hoog-energetische frames (zoals frames afkomstig van klinkers of van lettergrepen met een klemtoon) een veel grotere invloed hebben dan frames met lage energie (zoals frames afkomstig van fricatieven) wordt een normalisatie uitgevoerd op de kolommen van $\mathbf{V}$:

$$\bar{\mathbf{V}}_{ij} = \frac{\mathbf{V}_{ij}}{\sum_i \mathbf{V}_{ij}} \tag{4.1}$$

Merk op dat een dergelijke normalisatie de stilte-frames versterkt. Dit is een randeffect dat ongewenst is. De stilte-frames zullen namelijk door de normalisatie een grotere invloed hebben op de factorisatie. Er werd geopteerd om de stilte-frames toch in de matrix te laten staan aangezien deze ook deel uitmaken van het spraaksignaal. Een zelflerend algoritme moet namelijk robuust zijn om ook met frames zonder informatie-inhoud overweg te kunnen.

4.2.2 Matrix-factorisatie

De matrix $\mathbf{V}$ bestaat in dit geval uit 1.130.442 kolommen en 257 rijen, opgebouwd uit de trainingset van de TIMIT databank. Elke kolom beschrijft dus een vector in de 257-dimensionale ruimte. Merk op dat $\mathbf{V}$ geen negatieve getallen bevat. De factorisatie van $\mathbf{V}$ is dus mogelijk via NMF of het EM-algoritme. Via een matrix-factorisatie algoritme worden de matrices $\mathbf{W}$ en $\mathbf{H}$ berekend. De 257-dimensionale kolommen van $\mathbf{W}$ bevatten de basisvectoren die de nieuwe r-dimensionale ruimte opspannen. De matrix $\mathbf{H}$ bevat evenveel kolommen als de matrix $\mathbf{V}$. Deze kolommen bevatten de gewichten van de r basisvectoren om de overeenkomstige spectra van de spraakframes te reconstrueren. Er is dus een mapping gebeurd van de 257-dimensionale vectoren in de kolommen van $\mathbf{V}$ naar een r-dimensionale ruimte, waarbij voor r uiteraard een waarde kleiner dan 257 wordt gekozen. Dit is schematisch weergegeven in figuur 4.1.

[3]Dit wordt aangetoond in sectie 4.4.1 en sectie 4.5.1 voor het MSE-criterium, respectievelijk het divergentie-criterium.

4.2.3 Foneemherkenning

De r-dimensionale kolomvectoren van $\mathbf{H}$ zullen als kenmerkenvectoren gebruikt worden om aan foneemherkenning te doen via de ESAT-spraakherkenner. Eerst wordt de herkenner getraind met de vectoren in $\mathbf{H}$. De spraakherkenner zal hier eerst de eerste en tweede afgeleide toevoegen (zie sectie 2.2.1). Op deze manier hebben de kenmerkenvectoren een dimensie van $3r$. Deze dimensie wordt gereduceerd tot 36 via LDA (linear discriminant analysis). Dit algoritme selecteert automatisch de 36 nuttigste (discriminatieve) richtingen in de kenmerkenruimte[4].

Om de performantie van de herkenning te testen wordt met de testset uit de TIMIT databank een nieuwe matrix $\mathbf{V}$ opgesteld. Uit deze matrix wordt een nieuwe bijhorende matrix $\mathbf{H}$ berekend. Dit gebeurt via hetzelfde matrix-factorisatie algoritme, maar nu met de vaste matrix $\mathbf{W}$ die uit de trainingset werd bepaald. Het berekenen van $\mathbf{H}$ is een convex optimalisatieprobleem dat bijgevolg naar een gegarandeerd globaal optimum zal convergeren [12]. De kolommen van de gevonden matrix $\mathbf{H}$ bepalen de kenmerkenvectoren van de testset. Op deze vectoren wordt een foneemherkenning uitgevoerd, waarbij een PER (Phoneme-Error-Rate) zal berekend worden. Deze geeft een indicatie van de mogelijkheid van de kenmerkenset om de belangrijkste informatie uit spraaksignalen te extraheren[5].

4.3 Factorisatie via de singuliere-waardenontbinding

De eerste techniek die wordt toegepast om de matrix $\mathbf{V}$ te factoriseren is de singuliere-waardenontbinding, die in de ruimte met datapunten op zoek gaat naar de richtingen met de grootste variantie en bijgevolg ook de grootste informatie-inhoud. Deze richtingen worden gegeven door de vectoren horende bij de r grootste singuliere waarden. De data wordt daarom geprojecteerd op de ruimte opgespannen door de r linker singuliere vectoren, horende bij de r grootste singuliere waarden (merk op dat deze vectoren orthogonaal zijn). Op deze manier worden kenmerkenvectoren bekomen met een grote informatie-inhoud en een beperkte dimensie.

4.3.1 Voorverwerking

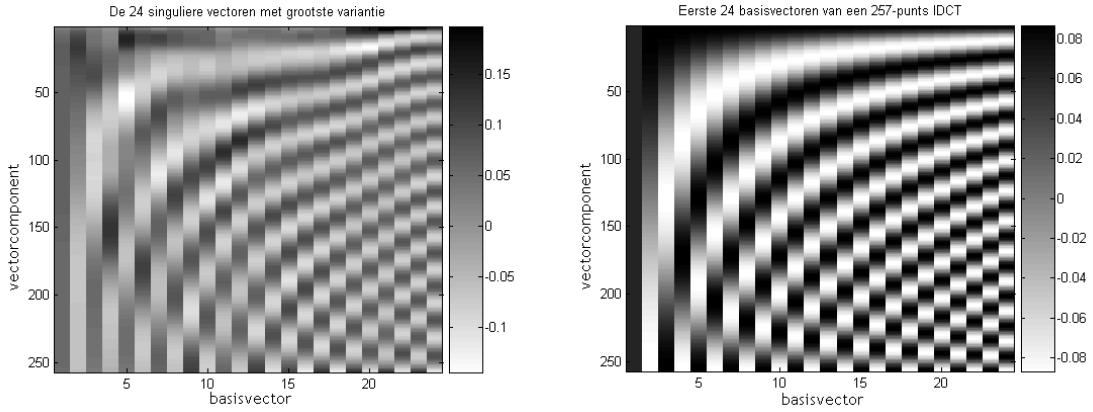
De volgende voorverwerking werd gebruikt voor het berekenen van de kolommen $\mathbf{v}_i$ van de matrix $\mathbf{V}$:

$$\mathbf{v}_i = \log\left(Env(|\mathbf{X}_i|)\right)$$

Waarbij $\mathbf{X_i}$ de FFT van frame $\mathbf{x_i}$ voorstelt. Merk op dat de logaritmische compressie al in de voorverwerking wordt toegepast. SWO minimaliseert namelijk een MSE-criterium. Indien $\mathbf{V}$ een groot dynamisch bereik heeft, is een dergelijk optimalisatieprobleem slecht geconditioneerd. In sectie 4.4.1 wordt hier dieper op ingegaan.

[4] Bij de gebruikte foneemherkenner geldt de nevenvoorwaarde van het afwezig zijn van foneemkennis niet meer helemaal. Het LDA algoritme gebruikt namelijk a-priorische kennis over fonemen voor de selectieprocedure van de 36 beste kenmerken. De kenmerken waaruit dit algoritme een selectie maakt zijn echter met de foneem-agnostische factorisatietechniek bepaald waardoor het begrip 'zelflerend' zijn betekenis behoudt.

[5] Het bepalen van de PER gebeurt uiteraard via supervised learning (cfr. het gebruik van LDA en HMM's). Dit is niet in tegenstrijd met de globale doelstelling van dit eindwerk. De PER dient namelijk enkel als maat voor de performantie van de kenmerkenset die via unsupervised learning werd gevonden.



(a) Singuliere Vectoren (b) IDCT

Figuur 4.2: Vergelijking tussen de gevonden basisvectoren via SWO (a) en de eerste 24 basisvectoren van een IDCT (b). De vectoren bevinden zich in de kolommen van de matrix. De lage frequenties bevinden zich bovenaan de matrix.

4.3.2 Resultaten

De factorisatie

Na de ontbinding worden uit de 257 singuliere waarden de r grootste geselecteerd. In dit experiment wordt $r = 24$ gekozen[6]. Deze 24 singuliere waarden verklaren 99.7% van de totale variantie van de data[7]. De bijhorende singuliere vectoren worden in figuur 4.2(a) weergegeven. Deze vectoren zijn orthogonaal en gelijken heel sterk op de basisvectoren van een (inverse) discrete cosinustransformatie (figuur 4.2(b)). Dit is niet verwonderlijk. De singuliere-waardenontbinding is namelijk een identieke factorisatie als de Principale Componenten Analyse (PCA), ook wel gekend als de Karhunen-Loève Transformatie (KLT). In het verleden werd al bewezen dat PCA een DCT benadert indien de dataset aan bepaalde voorwaarden voldoet (zie bv. [15]). DCT kan gezien worden als een data-onafhankelijke benadering van PCA, die net als PCA voor een energiecompactie en decorrelatie zorgt. Dit zijn twee belangrijke eigenschappen van een goede kenmerkenset.

Foneemherkenning

De projecties van de 257-dimensionale datavectoren in de 24-dimensionale ruimte opgespannen door de gevonden singuliere vectoren werden gebruikt als kenmerkenvectoren voor foneem-herkenning. In tabel 4.1 wordt de PER (phoneme-error-rate) van de testdata weergegeven voor een SWO met r=12 en r=24. Deze wordt vergeleken met de PER van de standaard foneemherkenner met de MEL-filterbank. Er werd een test uitgevoerd met zowel bigrammen als trigrammen[8]. De SWO met r=24 geeft herkenningsresultaten die slechts 0.3% slechter zijn dan de standaard foneemherkenner. Merk op

[6]Merk op dat 24 een arbitraire keuze is. In sectie 4.4.2 wordt deze keuze beter toegelicht.

[7]Met totale variantie wordt de som bedoeld van de varianties van alle 257 richtingen die door SWO werden gevonden.

[8]Dit slaat niet op het aantal toestanden die per HMM worden gebruikt, maar op het aantal fonemen die gecombineerd worden, afhankelijk van het gebruikte taalmodel. Bij bigrammen wordt gemodelleerd dat een foneem wordt beïnvloed door het voorgaande foneem. Bij trigrammen wordt een foneem beïnvloed door zowel het voorgaande als het volgende foneem. Dit laatste is nauwkeuriger en geeft normaal gezien betere resultaten.

Tabel 4.1: Vergelijking tussen PER voor MEL-filterbank en SWO

	SWO met 12 vectoren	SWO met 24 vectoren	MEL-filterbank
Bigrammen	30.47%	27.24%	26.96%
Trigrammen	29.24%	26.10%	25.84%

dat voor r=12 het LDA-algoritme geen invloed heeft, aangezien de kenmerkenvector al een dimensie heeft van 36.

4.3.3 Interpretatie van de resultaten

De factorisatie via SWO zal, rekening houdend met de opgelegde orthonormaliteitsvoorwaarden, gegarandeerd een globaal minimum vinden van de gemiddelde kwadratische fout. Het blijkt dat de gevonden basisvectoren spectrale eigenschappen hebben, gelijkaardig aan een IDCT. De spraakspectra worden gereconstrueerd aan de hand van de laagfrequente basisvectoren. Omdat de basisvectoren met hoge frequenties niet worden gebruikt, wordt een reconstructie met lagere resolutie bekomen.

De DCT wordt vaak gebruikt voor kenmerken-extractie in spraakherkenners. Aangezien SWO gelijkaardige basisvectoren vindt, is het niet verwonderlijk dat deze kenmerkenset ook goede herkenningsresultaten opleveren.

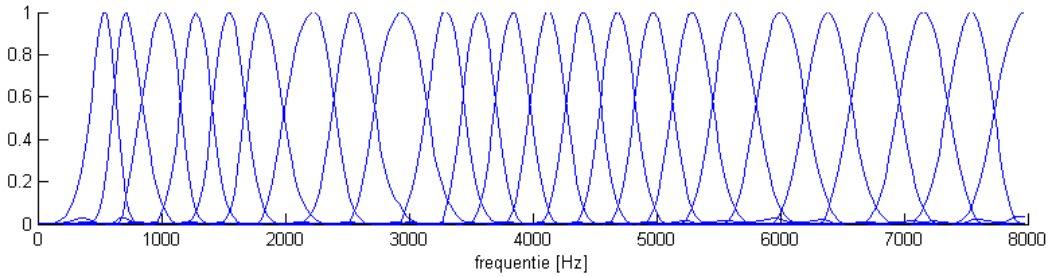
Ondanks het feit dat SWO optimaal is met betrekking tot de gemiddelde kwadratische fout, is de gevonden structuur moeilijk interpreteerbaar. De matrix $\mathbf{H}$ bevat namelijk negatieve gewichten. Dit betekent dat vectoren kunnen gebruikt worden om energie afkomstig van andere basisvectoren te annuleren. In secties 4.4 en 4.5 worden voorwaarden van niet-negativiteit opgelegd op deze gewichten.

4.4 Factorisatie met NMF volgens het MSE-criterium

Een factorisatie met het NMF-algoritme leidt tot een ontbinding zonder negatieve gewichten in de matrix $\mathbf{H}$. Op deze manier kunnen de basisvectoren in de matrix $\mathbf{W}$ als bouwblokken beschouwd worden, die additief worden samengesteld om de spectra te reconstrueren. De basisvectoren kunnen dus niet meer gebruikt worden om energie van andere vectoren te verwijderen. Deze sectie beschrijft de resultaten die werden bekomen via een ontbinding met het NMF-algoritme volgens het MSE-criterium.

4.4.1 MSE als optimalisatiecriterium

Het MSE-criterium (3.16) probeert de gemiddelde kwadratische fout van de reconstructie te minimaliseren. Een probleem met een dergelijk optimalisatiecriterium is slecht geconditioneerd voor data met een hoog dynamisch bereik. De kost is namelijk kwadratisch afhankelijk van de reconstructiefout. Concreet betekent dit dat het NMF-algoritme zich zal focussen op de hoogste pieken in de datamatrix $\mathbf{V}$. De basisvectoren zullen bepaald worden met het oog op een goede modellering van deze pieken. Een slechte reconstructie van deze pieken betekent namelijk dat door de kwadratische afhankelijkheid een heel hoge kost wordt aangerekend die de totale kost sterk domineert. Aangezien er voor elke frequentie een aantal frames bestaan met een hoog energetische piek, leidt een factorisatie van deze



Figuur 4.3: Een factorisatie volgens het MSE-criterium leidt tot een min of meer triviale oplossing. De resulterende basisvectoren bestaan uit allemaal ongeveer even brede pieken op verschillende plaatsen in het spectrum.

matrix tot een triviale oplossing (zie figuur 4.3). De bouwblokken in de matrix $\mathbf{W}$ bestaan dan uit 24 min of meer even smalle banden op verschillende locaties in het spectrum.

Om deze triviale oplossing te vermijden wordt, voorafgaand aan de factorisatie, de logaritme genomen[9] van de elementen in de matrix $\mathbf{V}$. Deze transformatie zorgt voor een compressie van het dynamisch bereik van de data in de matrix.

Het toepassen van een logaritmische transformatie heeft als nadeel dat de betekenis van de data in de matrix $\mathbf{V}$ moeilijker interpreteerbaar is. De formanten[10] die duidelijk aanwezig zijn in het vermogenspectrum zijn moeilijker te herkennen nadat de log-compressie werd toegepast. Vergelijk bv. de eerste en tweede figuur in figuur 4.8. De eerste figuur stelt een spraakspectrum na pitchverwijdering voor. De tweede figuur toont hetzelfde spraakspectrum na logaritmische compressie.

4.4.2 Keuze van het aantal basisvectoren

Een belangrijk nadeel van het NMF-algoritme is de a-priorische keuze van het aantal basisvectoren die mogen gebruikt worden bij de reconstructie. In SWO kan deze parameter na de ontbinding gekozen worden, gebaseerd op de variantie die in elke richting aanwezig is.

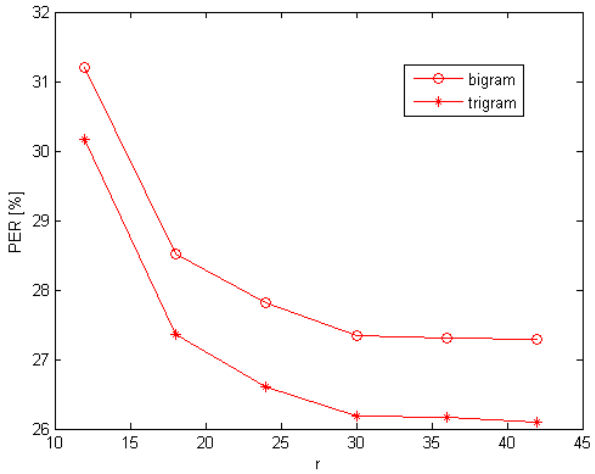
In het verleden werd empirisch vastgesteld dat de dimensie van kenmerkenvectoren voor spraakherkenning idealiter 20 à 30 bedraagt. Dit is vóór het toevoegen van de eerste en tweede afgeleide en de daarop volgende dimensiereductie door DCT of het LDA algoritme (zie sectie 2.2.1)[11]. Indien de oorspronkelijke kenmerkenvector minder dan 20 componenten bevat daalt de performantie van de herkenner significant.

Ook uit de resultaten van de experimenten blijkt deze vuistregel op te gaan. Er werden een aantal factorisaties uitgevoerd met het NMF algoritme volgens het MSE-criterium, voor verschillende keuzes van r. In figuur 4.4 wordt de PER getoond in functie van de gekozen dimensie r bij het gebruik van zowel trigrammen als bigrammen. De helling van beide curves wordt klein vanaf $r = 25$. Vanaf $r = 30$

[9] Bij niet-negatieve factorisaties, zoals in dit geval, wordt een bias van 1 opgeteld bij de elementen in $\mathbf{V}$. Dit zorgt ervoor dat alle elementen in de matrix groter zijn dan 1, zodat geen negatieve elementen verschijnen na het nemen van de logaritme.

[10] Formanten zijn pieken in het spectrum van het spraaksignaal. Deze ontstaan door resonanties in de trilholte gevormd door de keel-, neus- en mondholte. De plaats waar formanten optreden in het spectrum is afhankelijk van de vorm die deze holten op dat moment aannemen. De posities van formanten in het spectrum bevatten veel informatie omtrent de klank die werd geproduceerd.

[11] Indien geen MEL-filterbank gebruikt wordt, en rechtstreeks met cepstrale coëfficiënten van het oorspronkelijke spectrum gewerkt wordt, zijn 12 coëfficiënten voldoende.



Figuur 4.4: PER in functie van de dimensie r van de factorisatie met NMF volgens het MSE-criterium

is het toevoegen van extra componenten weinig zinvol.

Voor spraaksignalen die bemonsterd zijn aan 16 kHz wordt meestal een MEL-filterbank gebruikt met 24 banden voor het opstellen van de kenmerkenvectoren. Daarom wordt $r = 24$ gekozen in alle factorisaties die beschreven staan in dit hoofdstuk. Dit maakt het mogelijk om de herkenningsresultaten voor de 24 MEL-coëfficiënten te vergelijken met de herkenningsresultaten op basis van de kenmerken die gevonden werden via matrix-factorisatie.

4.4.3 Rekentijd en aantal iteraties

Experimenten tonen aan dat de kostfunctie met 20% gereduceerd wordt indien na 500 iteraties, nog 500 extra iteraties uitgevoerd worden. Bijgevolg werden voor elk experiment minimum 1000 iteraties uitgevoerd. In figuur 4.5 wordt het verloop van de kostfunctie weergegeven in functie van het aantal iteraties. Het verschil in PER bedraagt ongeveer 0.2% indien 1000 iteraties worden uitgevoerd in plaats van 500 iteraties.

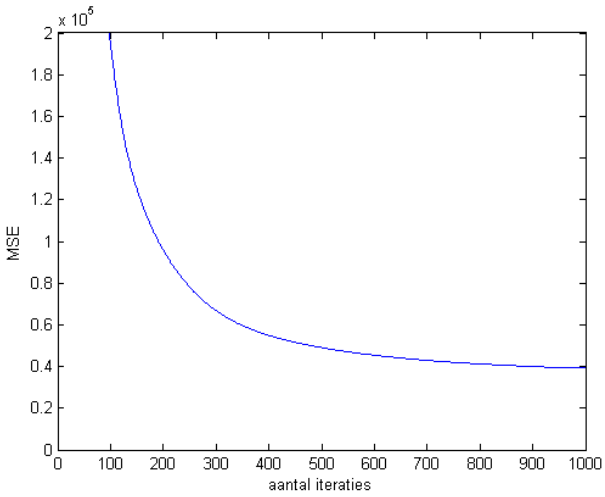
De rekentijd voor een matrix met de afmetingen van matrix $\mathbf{V}$ bedraagt ongeveer 40 seconden per iteratie op een 2,4 Ghz dual AMD opteron 280 processor met 16GB DDR400 geheugen. Een foneem-herkenningsexperiment op de TIMIT databank (inclusief training) duurt ongeveer 24 uur.

4.4.4 Voorverwerking

In dit experiment werden 2 verschillende voorverwerkingsmethoden getest:

1. $\mathbf{v}_i = \log\left(1 + Env(|\mathbf{X}_i|)\right)$
2. $\mathbf{v}_i = \log\left(1 + Smooth(|\mathbf{X}_i|^2)\right)$

Hierbij stelt $\mathbf{X_i}$ de FFT voor van frame x_i.



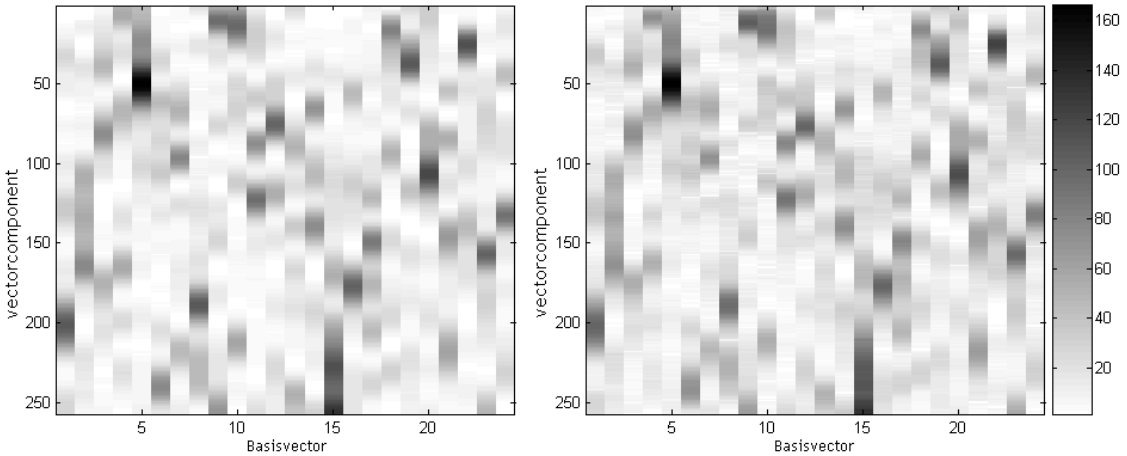
Figuur 4.5: MSE-kostfunctie in functie van het aantal uitgevoerde iteraties

4.4.5 Resultaten

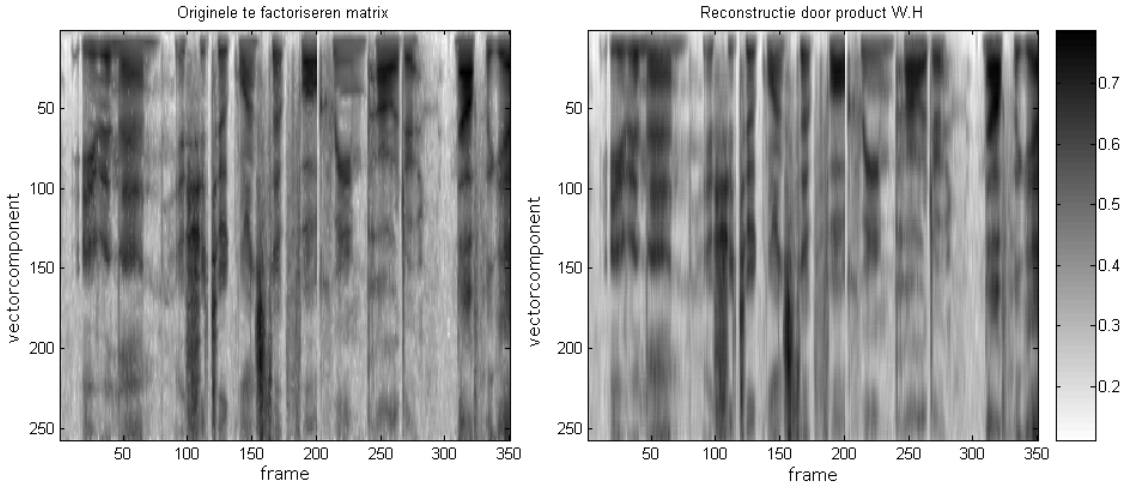
De factorisatie

Figuur 4.6 toont de matrix $\mathbf{W}$ die door het NMF algoritme werd berekend voor zowel voorverwerking 1 als 2, waarbij voor beiden dezelfde initialisatiematrices gebruikt werden. Het is duidelijk dat beide voorverwerkingsmethoden equivalente resultaten opleveren. De basisvectoren in de kolommen van deze matrices zijn moeilijk interpreteerbaar.

In figuur 4.7 worden een aantal kolommen van de te reconstrueren matrix $\mathbf{V}$ weergegeven, samen met de reconstructie door het product $\mathbf{WH}$. Het is duidelijk dat de factorisatie de oorspronkelijke matrix goed reconstrueert.

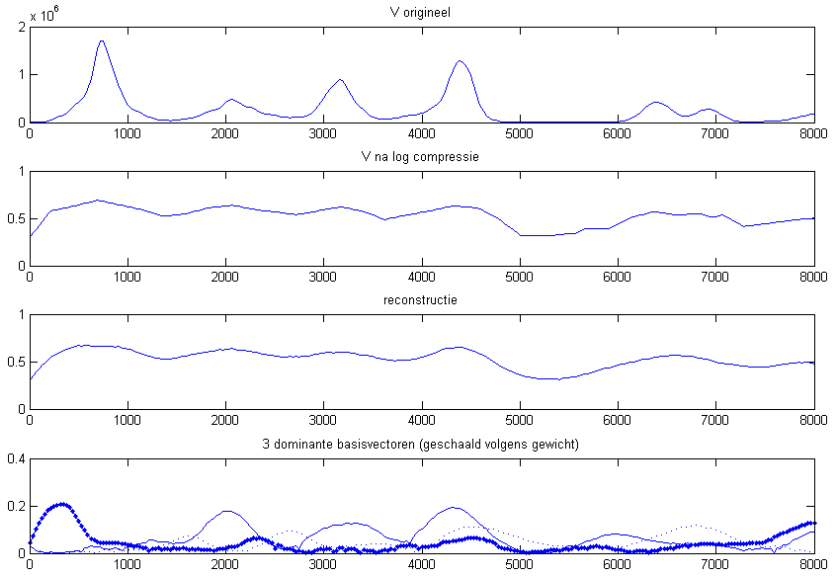


Figuur 4.6: Matrix $\mathbf{W}$, berekend door het NMF algoritme met MSE criterium voor voorverwerking 1 (links) en voorverwerking 2 (rechts). Voor beiden werden dezelfde initialisatiematrices gebruikt. De 24 basisvectoren die de nieuwe ruimte opspannen bevinden zich in de kolommen van deze matrix. De lage frequenties bevinden zich bovenaan de matrix.



Figuur 4.7: Een aantal kolommen van $\mathbf{V}$ (links) en de reconstructie ervan (rechts) door NMF met MSE-criterium.

In figuur 4.8 wordt de reconstructie van een kolom van de matrix $\mathbf{V}$ gedetailleerder weergegeven. In de bovenste figuur staat het oorspronkelijke spectrum van een spraakframe na het verwijderen van de pitch-harmonischen. De tweede figuur toont het overeenkomstige spectrum na log-compressie. De derde figuur toont de reconstructie via de basisvectoren in matrix $\mathbf{W}$. De onderste figuur tenslotte geeft de 3 basisvectoren uit de matrix $\mathbf{W}$ met het grootste gewicht in de overeenkomstige kolom van $\mathbf{H}$.



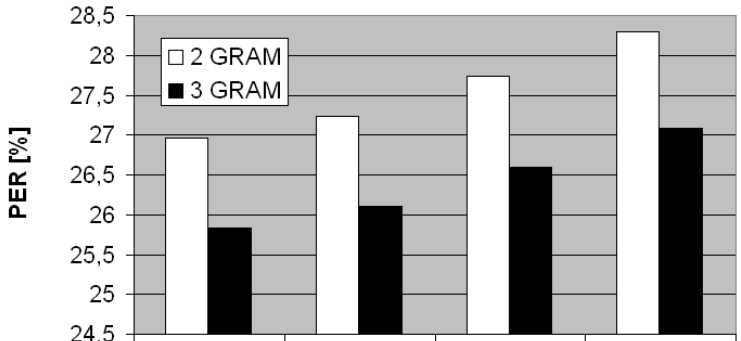
Figuur 4.8: Voorbeeld van de reconstructie van een kolom van de matrix $\mathbf{V}$.

Voor beide voorverwerkingsmethoden werden een aantal factorisaties uitgevoerd met telkens een andere intialisatie. Uit deze experimenten blijkt dat de basisvectoren die door het NMF algoritme gevonden worden heel sterk afhankelijk zijn van de initialisatie. Twee verschillende initialisaties geven een volledig verschillende factorisatie. Dit wijst erop dat NMF in lokale minima of zadelpunten blijft vastzitten.

Foneemherkenning

Voor elk factorisatie-experiment werd een foneemherkenning uitgevoerd. Hoewel elk experiment een verschillende set van basisvectoren oplevert, blijken deze sets ongeveer evenwaardig te zijn wat betreft PER. Het verschil in PER bedraagt maximum 0.6%. In figuur 4.9 wordt de PER van de NMF-kenmerken vergeleken met de PER van de SWO-kenmerken en de MEL-coëfficiënten. Er werden verschillende NMF experimenten uitgevoerd met verschillende initialisaties. In figuur 4.9 wordt enkel het experiment weergegeven dat het beste resultaat opleverde in teken van PER.

Er kan besloten worden dat NMF volgens het MSE-criterium goede kenmerkensets oplevert, die weliswaar steeds zwakkere PER resultaten opleveren in vergelijking met SWO en de MEL-filterbank. De voorverwerking met de Env-operator geeft aanleiding tot lagere PER in vergelijking met de $Smooth$-operator.



	Mel-filterbank	SWO	NMF (MSE) Env-operator	NMF (MSE) Smooth operator
□ 2 GRAM	26,96	27,24	27,74	28,3
■ 3 GRAM	25,84	26,1	26,6	27,08

Figuur 4.9: Vergelijking van de Phoneme-Error-Rates voor verschillende kenmerkensets

4.4.6 Interpretatie van de resultaten

De gevonden basisvectoren slagen erin om de oorspronkelijke matrix goed te reconstrueren. Dit blijkt ook uit de relatief lage phoneme-error-rates. De gewichten in de matrix $\mathbf{H}$ blijken dus redelijk goede kenmerken te zijn. Toch is dit resultaat niet bevredigend:

- De gevonden basisvectoren zijn moeilijk interpreteerbaar. Net zoals de SWO-vectoren hebben ze geen lokale eigenschappen. De meeste basisvectoren bevatten namelijk energie in meerdere gescheiden frequentiegebieden. Ze hebben bovendien geen spectrale betekenis, in tegenstelling tot de vectoren die werden gevonden via de singuliere-waardenontbinding[12].
- De factorisatie heeft door het toepassen van een logaritmische compressie een andere betekenis. Een additieve samenstelling van de basisvectoren betekent namelijk een multiplicatieve samenstelling van de vermogenspectra behorende bij deze basisvectoren. Een dergelijke interpretatie is complexer en minder intuïtief.

[12]DCT heeft ook een slechte lokaliteit, maar heeft een spectrale interpretatie. Een dimensiereductie betekent dat enkel de laagfrequente basisvectoren worden behouden. Hierdoor wordt de resolutie van de reconstructie lager.

- Het resultaat van het NMF-algoritme wijzigt indien een andere initialisatie wordt gekozen. Dit betekent dat de gebruikte kostfunctie veel lokale minima bevat.
- De phoneme-error-rates zijn hoger dan bij de gevonden kenmerken via de singuliere-waarden-ontbinding.

4.5 Factorisatie met NMF volgens het divergentie-criterium

In dit experiment wordt de matrix met spraakspectra gefactoriseerd met NMF volgens het divergentie-criterium. Dit probleem is beter geconditioneerd dan het MSE-criterium indien matrices met een groot dynamisch bereik worden gefactoriseerd. Dit laat toe om de logaritmische-compressie uit te stellen tot na de factorisatie.

4.5.1 Divergentie als optimalisatiecriterium

De kostfunctie van het divergentie-criterium voor een gegeven matrix $\mathbf{V}$ is gedefinieerd als

$$Div(\mathbf{V}||\mathbf{X}) = \sum_{i,j} \left(\mathbf{V}_{ij} \log \frac{\mathbf{V}_{ij}}{\mathbf{X}_{ij}} - \mathbf{V}_{ij} + \mathbf{X}_{ij} \right) \tag{4.2}$$

waarbij $\mathbf{X}$ de reconstructie van $\mathbf{V}$ is.

Stel dat een bepaald element v gereconstrueerd wordt als x, waarbij $\delta = \frac{x-v}{v}$ de relatieve fout is op de reconstructie. De kost voor de reconstructie van element v bedraagt in dit geval:

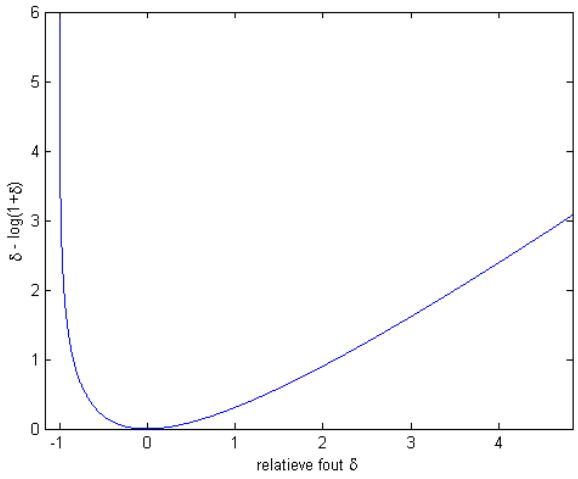
$$\begin{aligned} Div(v||x) &= v \log \left(\frac{v}{(1+\delta)v} \right) - v + (1+\delta)v \\ &= -v \log (1+\delta) + \delta v \\ &= (\delta - \log (1+\delta))\, v \end{aligned}$$

De factor $\delta - \log(1+\delta)$ zorgt ervoor dat positieve relatieve fouten aanleiding geven tot een kleinere kost dan negatieve relatieve fouten (zie figuur 4.10). Dit betekent dat de reconstructie van v eerder een overschatting zal zijn. Vermits de kost recht evenredig is met v zal de kost op een hoog-energetische waarde groter zijn voor een vaste relatieve fout. Bijgevolg zullen de hoog-energetische kolommen in de matrix $\mathbf{V}$ beter gemodelleerd worden dan de laag-energetische kolommen. Klanken met een grote energie-inhoud zullen dus zwaarder doorwegen bij het berekenen van de basisvectoren. In tegenstelling tot het MSE-criterium is dit echter een lineaire afhankelijkheid, zodat er minder dominantie is van een beperkt aantal grote pieken.

4.5.2 Rekentijd en aantal iteraties

In figuur 4.11 wordt het verloop van de kostfunctie weergegeven in functie van het aantal uitgevoerde iteraties. Voor de experimenten die in deze sectie worden besproken blijkt de kostfunctie reeds voldoende geconvergeerd te zijn na 200 iteraties[13]. De kostfunctie vermindert slechts met 2% indien

[13] Voor de experimenten waarbij in de voorverwerking een derde-machtswortel werd toegepast (zie verder), zijn 100 extra iteraties nodig voor voldoende convergentie.



Figuur 4.10: Divergentie-kost in functie van relatieve fout δ op een element met waarde 1. Een negatieve fout geeft een grotere kost dan een positieve fout. De eigenlijke kost moet nog vermenigvuldigd worden met de te reconstrueren waarde. Dit betekent dat een hogere kost wordt aangerekend voor grotere elementen in de matrix.

na 200 iteraties nog 800 extra iteraties worden uitgevoerd. Merk op dat er voor de NMF-experimenten met MSE-criterium (zie sectie 4.4) minimum 1000 iteraties nodig waren, aangezien de kostfunctie tussen de 500ste iteratie en de 1000ste iteratie nog met ongeveer 20% daalt. Uit de experimenten blijkt dat dit verschil niet te wijten is aan het verschil in optimalisatiecriterium, maar aan de logaritmische compressie.

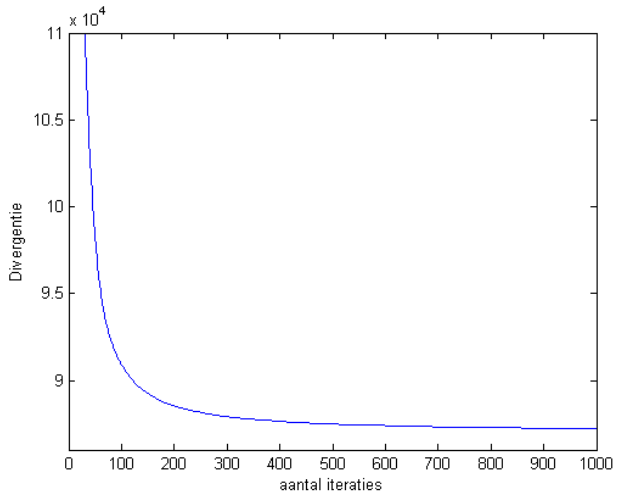
Voor convergentie van de getallen in matrices $\mathbf{W}$ en $\mathbf{H}$ blijken 200 iteraties echter niet voldoende te zijn. Zoals in sectie 3.2.5 werd aangehaald blijken de getallen in de matrices $\mathbf{W}$ en $\mathbf{H}$ veel trager te convergeren dan de kostfunctie. Indien de elementen van $\mathbf{H}$ als kenmerken gebruikt worden is het belangrijk dat deze getallen geconvergeerd zijn. Na 1000 iteraties veranderen de waarden van de elementen in $\mathbf{H}$ niet veel meer (gemiddeld met 0.01 %, in het slechtste geval met maximum 0.5%).

In de experimenten waarbij een PER werd bepaald, werden telkens 1000 iteraties uitgevoerd om de herkenning zo weinig mogelijk van convergentieproblemen te laten afhangen. Bovendien blijkt dat de PER nog met 1 à 2% kan gereduceerd worden, indien daarna nog 1000 extra iteraties worden uitgevoerd op de trainings-matrix $\mathbf{H}$, waarbij matrix $\mathbf{W}$ vast is.

Eenmaal de matrix $\mathbf{W}$ vast ligt kan in plaats van de NMF-updates een quasi-Newton-Raphson optimalisatie uitgevoerd worden op de rijen van matrix $\mathbf{H}$. Dit algoritme zorgt voor een snellere convergentie van de kostfunctie en laat bovendien de elementen van $\mathbf{H}$ veel sneller convergeren.

De rekentijd per iteratie bedraagt voor NMF met divergentie-criterium ongeveer 50 seconden op een 2,4GHz dual AMD opteron 280 processor met 16GB DDR400 geheugen. Er werd opnieuw gekozen om $r = 24$ te stellen[14].

[14]Er werd gecontroleerd of een hogere waarde voor r een significante verbetering geeft in termen van de PER op basis van de kenmerkenvectoren in matrix $\mathbf{H}$. Dit was niet het geval.



Figuur 4.11: Waarde van de kostfunctie (divergentie-criterium) in functie van het aantal uitgevoerde iteraties.

4.5.3 Voorverwerking

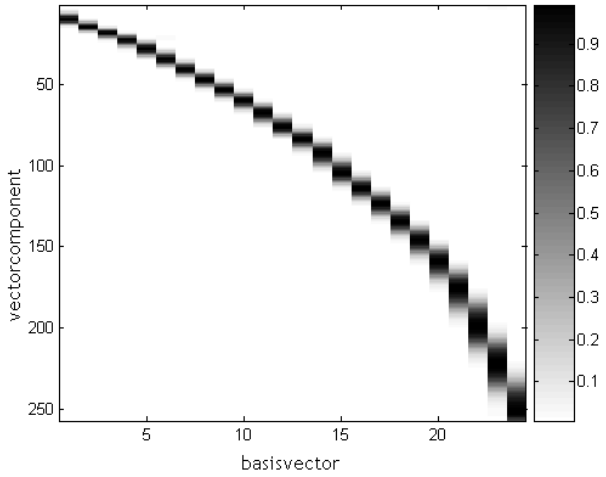
Aangezien het divergentie-criterium beter geschikt is dan het MSE-criterium voor matrices met een hoog dynamisch bereik, is het niet nodig om de logaritmische compressie al in de voorverwerking toe te passen[15]. De volgende voorverwerkingsmethoden werden onderzocht in dit experiment:

1. $\mathbf{v}_i = Env(|\mathbf{X_i}|^2)$
2. $\mathbf{v}_i = Smooth(|\mathbf{X_i}|^2)$
3. $\mathbf{v}_i = |\mathbf{X_i}|$
4. $\mathbf{v}_i = |\mathbf{X_i}|^2$
5. $\mathbf{v}_i = \sqrt[3]{Env(|\mathbf{X_i}|^2)}$

met $\mathbf{X_i}$ de FFT van frame x_i. Merk op dat de spectra na voorverwerking 3 of 4 nog pitch-informatie bevatten.

In de laatste voorverwerkingsmethode wordt een derde-machtswortel toegepast. Dit is geïnspireerd op de respons van de haarcellen in het oor, die mechanische trillingen van het basilair membraan omzetten in potentialen die naar de hersenen worden doorgestuurd. De respons van deze cellen voldoet aan een dergelijke wet [16]. Aangezien het toepassen van de derde-machtswortel het dynamisch bereik van de data verkleint, is een dergelijke voorverwerking voordelig in termen van de problemen die in de vorige sectie werden aangehaald.

[15]Indien voorafgaand aan de factorisatie toch de logaritme wordt genomen van de elementen in de matrix, wordt dezelfde matrix $\mathbf{W}$ bekomen als in figuur 4.6, indien dezelfde initialisatie wordt gebruikt. In dit geval is het divergentie-criterium dus min of meer equivalent met het MSE-criterium.



Figuur 4.12: Matrix $\mathbf{W}$ gevonden door NMF met divergentie-criterium. De kolommen bevatten de basisvectoren van de nieuwe ruimte. De lage frequenties bevinden zich bovenaan de matrix. De kolommen werden herschaald zodat hun maximum 1 bedraagt. De kolommen werden voor de duidelijkheid gepermuteerd.

4.5.4 Resultaten

De factorisatie

Bij gebruik van voorverwerkingsmethoden 1 tot en met 4 levert het NMF algoritme nagenoeg steeds dezelfde basisvectoren. Figuur 4.12 toont de matrix $\mathbf{W}$ voor voorverwerking 1. In figuur 4.13 worden al de kolommen van $\mathbf{W}$ samen in het frequentiespectrum weergegeven. Het is opmerkelijk dat deze figuur sterk gelijkt op de MEL-filterbank, die in sectie 2.3 werd geïntroduceerd (vergelijk met figuur 2.4). De banden worden breder op hogere frequenties. Deze gelijkenis wordt in sectie 4.5.6 uitvoerig behandeld.

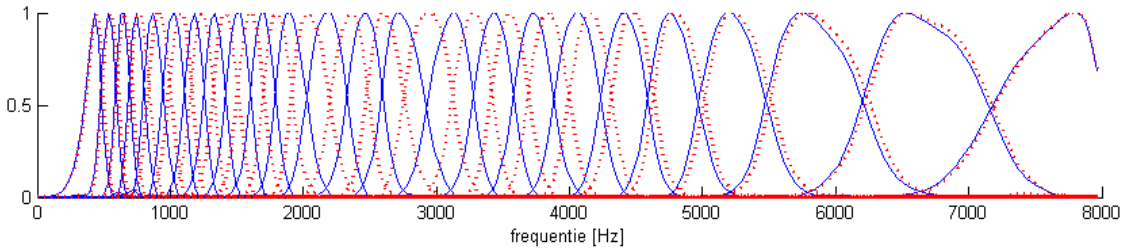
Verschillende initialisaties van het NMF-algoritme geven telkens qua vorm ongeveer dezelfde basisvectoren, met als enige verschil een al dan niet beperkte verschuiving van de banden in het frequentiespectrum (stippellijn in figuur 4.13). De overeenkomstige waarden van de kostfunctie zijn maximaal 2,5% hoger dan de laagst bereikte kost.

De basisvectoren die in stippellijn worden weergegeven in figuur 4.13 zijn afkomstig van een factorisatie waarbij de kostfunctie 1601 bedraagt. Voor de volle lijn bedroeg deze 1622. In beide gevallen bleek de kostfunctie geconvergeerd te zijn. In teken van divergentie geeft de volle lijn dus aanleiding tot een minder goede reconstructie[16]. De PER via de basisvectoren in volle lijn is echter een stuk lager dan de basisvectoren in stippellijn. Ook uit andere experimenten blijkt dat een lagere kostfunctie geen garantie is voor een lagere phoneme-error-rate. Bijgevolg werd voor elk experiment een foneemherkenning uitgevoerd, onafhankelijk van de waarde van de kostfunctie.

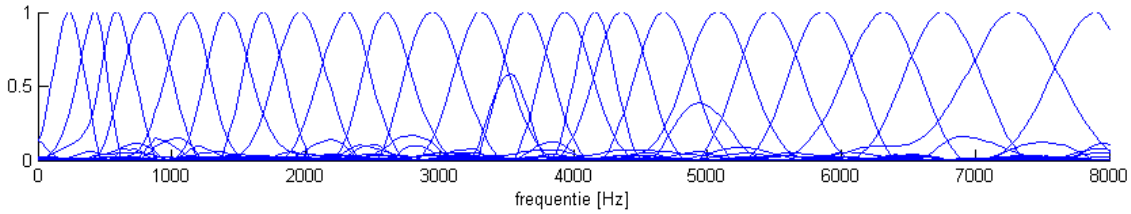
De eerste vier voorverwerkingsmethoden geven allen vergelijkbare basisvectoren (met eventuele translaties in het spectrum). Dit betekent dat de aanwezigheid van pitch[17], of de methode waarmee

[16] Merk op dat de waarde van de kostfunctie niet enkel afhankelijk is van de basisvectoren in $\mathbf{W}$, maar ook van de elementen in de matrix $\mathbf{H}$.

[17] Indien nog pitch-informatie aanwezig is bevatten de basisvectoren weliswaar iets hogere zijlobben. Het effect hiervan op de PER is verwaarloosbaar.



Figuur 4.13: Geschaalde basisvectoren van twee factorisaties met verschillende initialisatie. De kostfunctie van de factorisatie heeft een kostfunctie van 1622. Voor de basisvectoren in stippellijn bedraagt de kost 1601.



Figuur 4.14: De geschaalde basisvectoren na factorisatie met voorverwerkingsmethode 5 (derde machts-wortel). De pieken met een amplitude kleiner dan 1 zijn zijlobben.

pitch werd verwijderd, weinig invloed heeft op de resulterende factorisatie. Indien de pitch niet verwijderd wordt in de voorverwerking verdwijnt deze bij reconstructie door uitmiddeling.

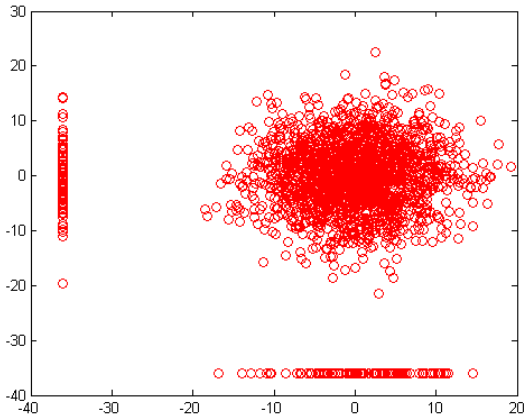
Ook voor de voorverwerking met een derde-machtswortel (voorverwerkingsmethode 5) wordt een stijging in de breedte van de banden waargenomen in functie van de frequentie. De overeenkomstige basisvectoren worden in figuur 4.14 weergegeven. De banden op de lage frequenties zijn over het algemeen iets breder dan in de andere experimenten. Bijgevolg zijn er meer banden beschikbaar in de hoge frequenties. Dit is vermoedelijk te wijten aan de compressie van het dynamisch bereik door de derde-machtswortel. Hierdoor winnen de hoge frequenties in energie ten opzichte van de lage frequenties. Door de energie-afhankelijkheid van het divergentie-criterium wordt de resolutie bij de reconstructie van de hoge frequenties verhoogd. De zijlobben zijn beduidend hoger dan in de andere vier voorverwerkings-methoden. Hiervoor werd geen verklaring gevonden.

Problemen bij de foneemherkenning

Hoewel de basisvectoren van de NMF-factorisatie met divergentie-criterium sterk gelijken op de MEL-filterbank, behaalt de beste kenmerkenset van deze factorisatie een hoge PER van 31,7% bij het gebruik van trigrammen. Dit is meer dan 5% hoger dan bij het gebruik van de MEL-filterbank. Bovendien was er een grote spreiding tussen de phoneme-error-rates van de verschillende experimenten (afhankelijk van de willekeurige initialisatie en de gebruikte voorverwerking). Het verschil tussen het beste en het slechtste resultaat bedroeg 3,2%[18].

Deze hoge phoneme-error-rates blijken het gevolg te zijn van de grote hoeveelheid nullen in de matrix $\mathbf{H}$ (gemiddeld 2,4% van de elementen). Na een logaritmische compressie liggen bijna alle elementen

[18]Deze experimenten verschilden enkel in initialisatiematrices. Ook tussen de gebruikte voorverwerkingsmethoden zijn er grote verschillen in PER. In het vervolg van deze sectie wordt aangetoond wat deze grote verschillen veroorzaakt.



Figuur 4.15: De grote hoeveelheid nullen in de matrix $\mathbf{H}$ zorgen voor artefacten in de kenmerken-ruimte. Deze artefacten kunnen niet via een gaussiaanse kansdichtheidsfunctie gemodelleerd worden.

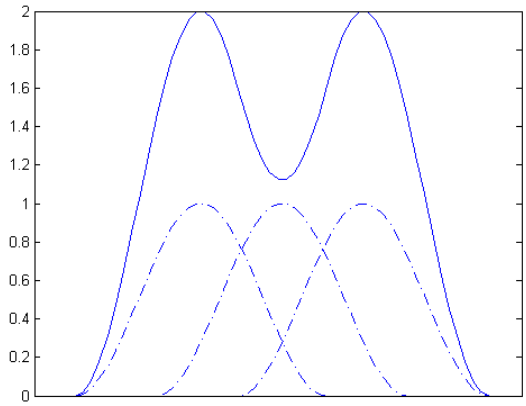
van de matrix $\mathbf{H}$ binnen het interval $[-10, 10]$. De nullen daarentegen worden door het nemen van een logaritme allemaal op de waarde -36 gemapt[19]. Dit zorgt voor artefacten in de kenmerken-ruimte. In figuur 4.15 wordt dit effect geïllustreerd in een 2-dimensionale ruimte voor een fictieve matrix $\mathbf{H}$ met 2 rijen. Elke kolom van $\mathbf{H}$ is hier een 2-dimensionale vector die als een punt in het vlak wordt voorgesteld. De kolommen die een nul bevatten veroorzaken artefacten in deze figuur (cfr. de horizontale en verticale lijn met collineaire punten). Zoals in sectie 2.2.2 werd uitgelegd, worden er tijdens de training van de spraakherkenner Gaussiaanse kansdichtheidsfuncties geschat, die de punten in de kenmerken-ruimte zo goed mogelijk verklaren. De artefacten in de kenmerken-ruimte, gevormd door de nullen in de matrix, hebben een variantie van nul in één richting, en passen daarom niet onder een Gaussiaanse kansverdeling. Dit zorgt voor vreemde effecten bij het schatten van de parameters van de Gaussianen tijdens de training, wat nefast is voor de herkenningsresultaten.

Het feit dat deze nullen verschijnen in de matrix $\mathbf{H}$ is een inherent probleem aan het maken van additieve lineaire combinaties met basisvectoren met lokale eigenschappen. Een element h_{ij} met waarde nul in matrix $\mathbf{H}$ betekent dat de bijhorende basisvector $\mathbf{w}_i$ niet gebruikt wordt in de reconstructie van kolom $\mathbf{v}_j$. Dit betekent dat de energie in kolom $\mathbf{v}_j$ op de plaats waar de energie van basisvector $\mathbf{w}_i$ gelokaliseerd is, reeds volledig verklaard wordt door andere basisvectoren. Deze situatie is weergegeven in figuur 4.16. Het toewijzen van een strikt positief gewicht aan basisvector $\mathbf{w}_i$, zal de reconstructiefout op die plaats alleen maar vergroten.

Door het verhogen van de bias ϵ bij het nemen van de logaritme zullen de artefacten in de kenmerken-ruimte verschuiven naar de puntenwolk waar de relevante punten zich bevinden. Dit lost het probleem gedeeltelijk op, aangezien de punten met een nul in de overeenkomstige kolom van $\mathbf{H}$ geen outliers meer zijn. Toch is dit geen goede oplossing omdat er zich in de puntenwolk nog steeds artefacten bevinden van collineaire punten. Het blijft problematisch om goede parameters te schatten voor de Gaussianen die dit moeten modelleren.

Om het probleem van de nullen in $\mathbf{H}$ volledig op te lossen kan eventueel laag-energetische additieve ruis toegevoegd worden. Dit zorgt ervoor dat er geen artefacten meer zijn met oneindig smalle breedte. Er werd echter voor een iets elegantere oplossing gekozen. Alvorens de herkenner te trainen wordt op

[19] Aangezien $\log(0) = -\infty$ wordt een klein getal ϵ opgeteld bij de waarden in matrix $\mathbf{H}$. Voor $\epsilon = 2,2\ 10^{-16}$ wordt dit $\log(0 + \epsilon) = -36$.



Figuur 4.16: Twee naburige pieken (volle lijn) moeten met drie lokale basisvectoren (stippellijn) gereconstrueerd worden. De twee pieken kunnen door de eerste en derde basisvector gemodelleerd worden. Aangezien dit reeds voldoende energie verklaart op de plaats van de middelste basisvector, zal deze niet gebruikt worden voor de reconstructie. Bijgevolg krijgt deze gewicht nul.

de matrix $\mathbf{H}$ de volgende bewerking toegepast:

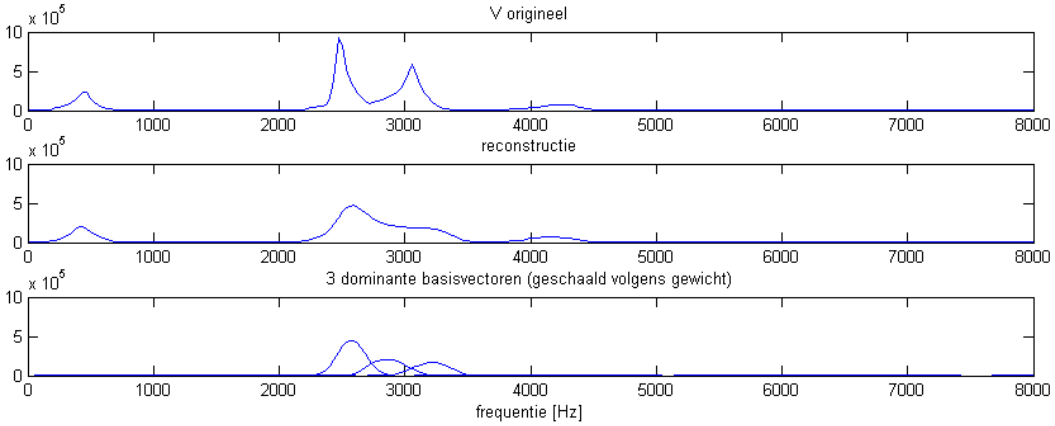
$$\hat{\mathbf{H}} = (1 - \alpha)\mathbf{H} + \alpha\mathbf{W}^T\mathbf{V} \tag{4.3}$$

met α een klein getal. Dit komt neer op het toevoegen van additieve ruis die sterk gecorreleerd is met de getallen in $\mathbf{H}$. Bemerk de analogie met de initialisatie van $\mathbf{H}$, die in sectie 3.4 werd uitgelegd.

Resultaten van de foneemherkenning

Het verwijderen van de nullen in $\mathbf{H}$ volgens (4.3) met $\alpha = 0.02$ zorgt voor een daling van gemiddeld 4% in PER. Toch is het beste resultaat in PER nog steeds 1 à 2 % slechter dan de standaard-herkenner met MEL-filterbank. Dit is niet volledig te wijten aan de trage convergentie van de getallen in matrix $\mathbf{H}$. De belangrijkste reden is het feit dat het maken van lineaire combinaties van de basisvectoren in $\mathbf{W}$ volgens het divergentie-criterium de energie niet bewaart. In bepaalde gevallen zal het NMF-algoritme de getallen in $\mathbf{H}$ zo kiezen dat bepaalde pieken 'opzettelijk' slecht gemodelleerd worden. De reden hiervoor is om de totale kost afkomstig van de fouten op de naburige frequenties te reduceren. Indien de pieken tussen twee opeenvolgende banden van de basisvectoren gelegen zijn, blijkt een slechte modellering dus een nettowinst op te leveren in de totale kost. Dit wordt geïllustreerd in figuur 4.17, die de reconstructie toont van een frame uit de matrix $\mathbf{V}$. De piek op 3000 Hz wordt hier slecht gereconstrueerd. De reden hiervoor is dat deze piek net tussen 2 basisvectoren ligt. Een goede reconstructie van de top van de piek zal bijgevolg grote kosten veroorzaken in de naburige frequentiepunten. NMF kiest daarom om deze piek niet te modelleren, waardoor de energie in dit frequentiegebied onderschat wordt[20]. Merk op dat het onderschatten van de pieken in deze figuur in strijd lijkt te zijn met de eerdere bewering dat het divergentie-criterium eerder overschattingen maakt. Dit is te verklaren door het feit dat hier een smalle piek gereconstrueerd moet worden door middel van een relatief brede band. De winst op de kostfunctie door het overschatten van deze piek weegt niet op tegen het grote aantal fouten die in een breed gebied rond de piek worden gemaakt.

[20]Merk op dat deze figuur een extreem geval is. In de meeste frames is de reconstructie van de formanten nauwkeuriger.

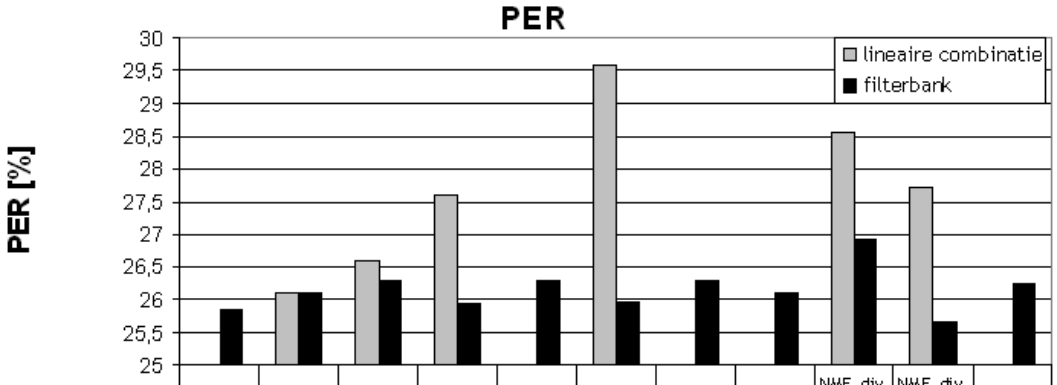


Figuur 4.17: Reconstructie van een frame uit $\mathbf{V}$. De bovenste figuur geeft het originele frame. In het midden staat de reconstructie. De onderste figuur toont de drie belangrijkste basisvectoren die voor de reconstructie werden gebruikt.

Het blijkt dat het toepassen van de gevonden matrix $\mathbf{W}$ als filterbank veel robuuster is dan het gebruik van deze matrix in additieve lineaire combinaties. Dit komt neer op een waarde 1 voor α in (4.3). Hierdoor wordt 1 à 2 % winst in PER behaald. In figuur 4.18 worden deze PER resultaten samengevat. Enkel de resultaten voor het gebruik van trigrammen worden weergegeven. Indien meerdere experimenten werden uitgevoerd, wordt in figuur 4.18 enkel het beste resultaat weergegeven[21]. De betekenis van de labels van de verschillende categorieën is respectievelijk:

- **MEL-filterbank**: De standaard-herkenner die gebruik maakt van de MEL-filterbank.
- **SWO**: Factorisatie met SWO. Merk op dat een lineaire combinatie van de basisvectoren in dit geval identiek is aan het gebruik van de basisvectoren als een filterbank, wegens de orthogonaliteit van deze vectoren.
- **NMF_mse**: NMF met MSE-criterium. Dit is steeds met voorafgaande logaritmische compressie (zowel bij het gebruik als filterbank, als voor het maken van lineaire combinaties).
- **NMF_div preproc 1 tot en met 5**: NMF met divergentie-criterium, voor de respectievelijke voorverwerkingsmethoden die in sectie 4.5.3 in een genummerde lijst werden opgesomd. De volgorde is dezelfde als in deze lijst.
- **NMF_div no pre-emp**: NMF met divergentie-criterium zonder pre-emphasis in de voorverwerking. Het globale effect op de kolommen van $\mathbf{W}$ is een groter aantal banden in de lage frequenties. Boven 4000 Hz worden nu slechts 3 banden gevonden.
- **NMF_div lower bands**: NMF met divergentie-criterium, waarbij werd opgelegd dat er 2 banden in de frequenties tussen 0 en 300 Hz moeten aanwezig zijn. Dit is mogelijk door in 2 kolommen van de initialisatiematrix $\mathbf{W}_0$ nullen te zetten op alle andere frequenties. De update-formules van NMF zijn zodanig dat een nul steeds een nul blijft. De bedoeling is om de twee eerste banden uit de MEL-filterbank ook via NMF te verkrijgen. Deze ontbreken namelijk in de standaard NMF-factorisatie (deze observatie wordt uitvoeriger besproken in sectie 4.5.6). Er wordt een hogere kostfunctie bekomen dan in het geval van 'NMF_div preproc 1'. Toch wordt er ongeveer 0.3% gewonnen op vlak van PER.
- **NMF_div balanced**: NMF met divergentie-criterium met scalering van $\mathbf{V}$ om foneem-aantallen te balanceren. Hoewel de TIMIT databank een van de meest gebalanceerde databanken is qua

[21] De spreiding tussen de PER is verwaarloosbaar bij het gebruik van $\mathbf{W}$ als filterbank.



	MEL-filterbank	SWO	NMF_mse	NMF_div preproc 1	NMF_div preproc 2	NMF_div preproc 3	NMF_div preproc 4	NMF_div preproc5	NMF_div no pre-emp	NMF_div lower bands	NMF_div balanced
lineaire combinatie		26,1	26,6	27,6		29,58			28,56	27,71	
filterbank	25,84	26,1	26,3	25,95	26,28	25,97	26,3	26,1	26,92	25,68	26,24

Figuur 4.18: PER resultaten voor verschillende factorisaties. Ontbrekende resultaten in de tabel duiden op het feit dat het experiment niet werd uitgevoerd. Indien meerdere experimenten werden uitgevoerd, wordt het beste resultaat weergegeven. De lichte balken gelden voor een herkenning via de elementen in matrix $\mathbf{H}$, de zwarte balken gelden voor het gebruik van $\mathbf{W}$ als filterbank.

foneeminhoud, geldt er dat bepaalde fonemen veel frequenter voorkomen dan anderen. Dit zou een eventuele bias kunnen veroorzaken op de factorisatie. Om dit te vermijden werd elke kolom $\mathbf{v}_i$ van $\mathbf{V}$ in dit experiment gewogen met $\frac{1}{n_i}$ waarbij n_i het aantal uitingen voorstelt van het corresponderende foneem in de TIMIT databank[22]. Aangezien het divergentie-criterium lineair afhankelijk is van de grootte van de elementen in $\mathbf{V}$ wordt er op deze manier gecompenseerd voor het niet gebalanceerd zijn van de databank. Aangezien de testset op dezelfde manier als de trainingset ongebalanceerd is, heeft het balanceren van de trainingset een negatieve invloed op de PER.

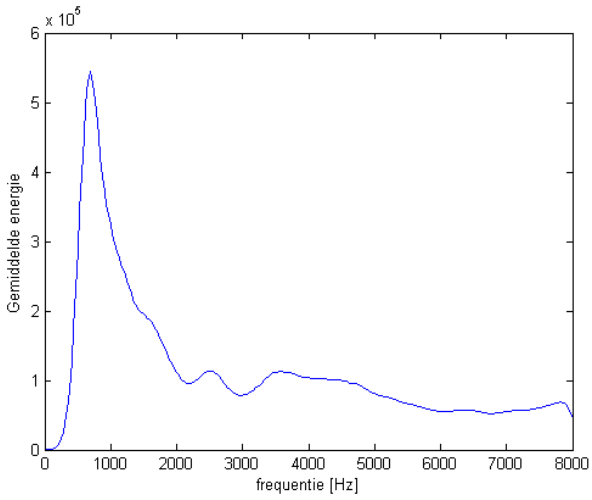
De conclusie is dat de gevonden factorisatie volgens het divergentie-criterium een kenmerkenset vindt die het even goed doet als de standaard-herkenner. Met een beetje hulp door het opleggen van 2 banden in de laagste frequenties, wordt zelfs een lagere PER bekomen. Deze verbetering is echter niet significant.

4.5.5 Interpretatie van de resultaten

Uit de resultaten blijkt dat de vier problemen opgesomd in sectie 4.4.6 worden opgelost indien het divergentie-criterium wordt gebruikt in plaats van het MSE-criterium:

- De basisvectoren zijn makkelijk interpreteerbaar en hebben lokale eigenschappen.
- De data in matrix $\mathbf{V}$ bestaat nog steeds uit onvervormde spraakspectra, zonder compressie van het dynamisch bereik door een logaritmische transformatie.
- Verschillende initialisaties geven steeds een gelijkaardige set basisvectoren. Er zijn slechts kleine verschillen (zoals kleine translaties van bepaalde banden in het spectrum).

[22] Merk op dat in dit geval een lichte vorm van a-priorische foneemkennis wordt gebruikt.



Figuur 4.19: De gemiddelde energie in de matrix $\mathbf{V}$ in functie van frequentie

- Er werd een kenmerkenset gevonden die minstens even goed is als die van een standaard foneemherkenner. Indien het factorisatie-algoritme verplicht wordt om ook banden in de allerlaagste frequenties te zetten, wordt zelfs een iets betere PER bereikt.

De gelijkenis tussen de gevonden factorisatie en de MEL-filterbank is opmerkelijk. In de volgende sectie wordt deze gelijkenis uitgebreider onderzocht.

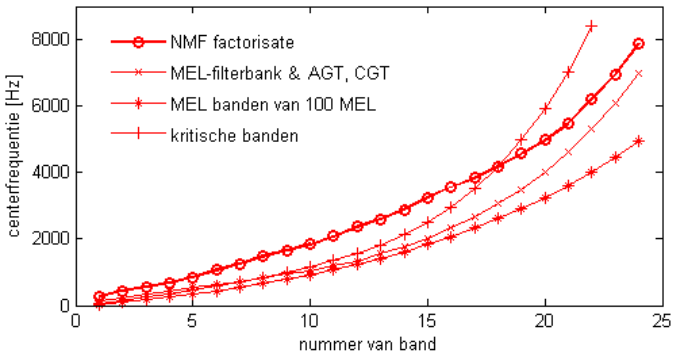
4.5.6 Vergelijking met de MEL-filterbank en andere gehoormodellen

Bij de factorisatie van $\mathbf{V}$ met NMF volgens het divergentie-criterium wordt een kenmerkenset gevonden die opvallend goed gelijkt op de MEL-filterbank. In deze sectie wordt deze gelijkenis geanalyseerd. Er wordt ook een vergelijking gemaakt met andere gehoormodellen die in de literatuur beschreven zijn.

Ontbrekende banden

Een eerste verschil tussen de gevonden basisvectoren en de MEL-filterbank is het ontbreken van de banden op de laagste frequenties (vergelijk figuur 4.13 met figuur 2.4). Er werden experimenten uitgevoerd waarbij NMF verplicht werd om deze frequenties te modelleren. Dit is mogelijk door in enkele kolommen van de initialisatiematrix $\mathbf{W}_0$ niet-nul waarden te zetten op deze frequenties en nullen op alle andere frequenties. De update-formules van NMF zijn zodanig dat een nul steeds een nul blijft. In elk van deze experimenten was de kost steeds hoger dan de experimenten waarbij deze banden niet werden gevonden. Dit wijst erop dat het ontbreken van deze banden niet het gevolg is van lokale minima of convergentieproblemen.

De reden voor het ontbreken van de laagfrequente banden is het feit dat de gemiddelde energie op deze frequenties quasi nul is (zie figuur 4.19). Dit is enerzijds wegens het toepassen van pre-emphasis en anderzijds omdat de data in de TIMIT-databank op de allerlaagste frequenties bijna geen energie bevat. Aangezien het divergentie-criterium lineair afhankelijk is van de energie, is het logisch dat NMF geen



Figuur 4.20: Centerfrequenties van de NMF-basisvectoren en van enkele filterbanken gebaseerd op verschillende gehoormodellen. De curve van de NMF-basisvectoren wordt met een dikkere lijn weergegeven.

aandacht schenkt aan deze frequenties. Indien geen pre-emphasis wordt toegepast, vindt NMF wel banden op deze frequenties.

De spraakdata in de TIMIT databank is gefilterd met een hoogdoorlaatfilter om de DC-component te verwijderen. Hetzelfde algoritme werd uitgetest op de Resource Management databank[23]. Deze databank werd niet met een hoogdoorlaatfilter gefilterd. NMF vindt op deze databank wel een band op de laagste frequenties (mits een artefact op 0 Hz, vermoedelijk door een DC-component op de spraakdata).

Verdeling van de banden

In figuur 4.20 worden de centerfrequenties van de banden van de NMF-basisvectoren vergeleken met de centerfrequenties van andere gehoormodellen: de MEL-filterbank (Davis & Mermelstein benadering), de analytische en conventionele gamma-tone filterbank (AGT en CGT) [18], de kritische banden op basis van de bark-schaal, en een MEL-filterbank met niet-overlappende banden van 100 MEL (dit is op basis van de echte MEL-schaal).

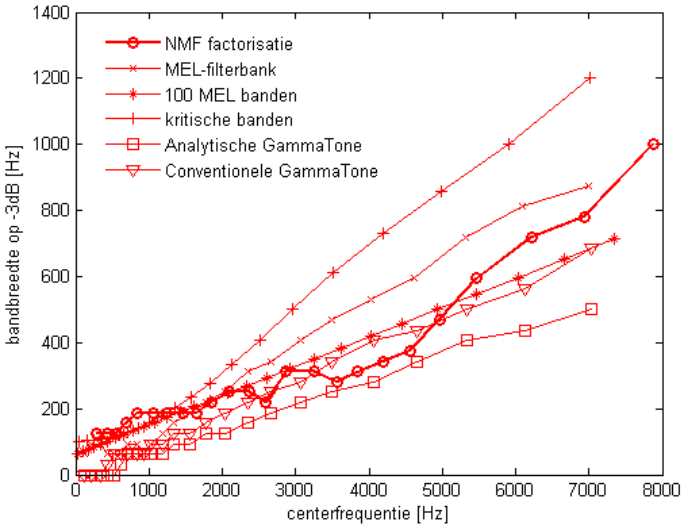
De curve voor de NMF-basisvectoren ligt hoger dan de curves van de gehoormodellen. De vorm is echter gelijkaardig aan de andere curves. Merk op dat de NMF-curve een bias heeft door het ontbreken van de eerste twee banden, zoals reeds eerder werd aangegeven.

Breedte van de banden

Figuur 4.21 toont de -3dB bandbreedte van de banden van de NMF-basisvectoren en van de gehoormodellen die hierboven werden opgesomd.

De curve van de NMF-basisvectoren ligt tussen de curves van de verschillende gehoormodellen, maar er is geen gehoormodel waarmee er een grote gelijkenis is. De curves van de gehoormodellen zijn over het algemeen redelijk lineair. De bandbreedtes van de NMF-basisvectoren verlopen lineair met een kleine helling tot ongeveer 4500 Hz. Vanaf 4500 Hz is de stijging in bandbreedte veel groter en stijgt de curve lineair met een grotere helling. Deze discontinuïteit op 4500 Hz werd in alle experimenten

[23] Meer informatie omtrent deze databank is te vinden in [17].



Figuur 4.21: -3 dB bandbreedte van de verschillende banden van de NMF basisvectoren (dikke lijn) en van enkele gehoormodellen.

(met verschillende voorverwerkingsmethoden) waargenomen. Dit is niet verwonderlijk: het is een bekend feit dat menselijke spraak vooral informatie bevat in de frequenties tot 4000 Hz. Buiten deze frequenties worden geen echte formanten meer waargenomen en is het spectrum turbulenter (vooral in het geval van fricatieven[24]).

4.6 De link tussen productie en analyse van spraak

4.6.1 Productie en analyse op elkaar afgestemd

De factorisatie volgens het divergentie-criterium levert een kenmerkenset op die gelijkenissen vertoont met de MEL-filterbank. Deze filterbank wordt vaak gebruikt voor kenmerken-extractie en is gebaseerd op een gehoormodel dat de frequentieresolutie in het menselijk auditief systeem modelleert. De NMF-factorisatie daarentegen gebeurt op basis van de analyse van spraaksignalen. Het feit dat beiden een gelijkaardige kenmerkenset opleveren is opmerkelijk. Op basis van deze observatie kan de stelling geponeerd worden dat, gedurende de evolutie van de mensheid, de productie en analyse van spraaksignalen heel goed op elkaar afgestemd zijn om de informatie-overdracht zo efficiënt mogelijk te maken. De productie van menselijke spraak is m.a.w. zodanig dat het gehoorsysteem de informatie, die nodig is voor de herkenning van de spraak, zo goed mogelijk kan extraheren.

4.6.2 Correctheid van de factorisatie

Om de bovenstaande stelling te onderzoeken wordt nagegaan of het divergentie-criterium leidt tot een factorisatie op basis van structuur in $\mathbf{V}$ of enkel op basis van de energieverdeling in $\mathbf{V}$. Zoals in sectie

[24] Dit is bv. de ‘s’ klank en de ‘f’ klank.

4.5.1 werd aangetoond heeft het divergentie-criterium namelijk een voorkeur om hoog-energetische delen in de matrix $\mathbf{V}$ goed te modelleren[25]. Het is een feit dat de lage frequenties in spraaksignalen veel meer energie bevatten.

Een observatie van de spectra in de matrix $\mathbf{V}$ toont aan dat er over het algemeen inderdaad een verschil is in de gemiddelde hoogte van pieken op hoge en lage frequenties. De grootste pieken zijn echter ongeveer even hoog in alle frequentiegebieden. Dit is mede dankzij een pre-emphasis en de normalisatie (4.1). De gemiddelde energie is in de laag-frequente gebieden echter een stuk hoger dan in de hoog-frequente gebieden (zie figuur 4.19), omdat op lage frequenties meer hoge pieken voorkomen dan op hogere frequenties.

Het feit dat NMF smalle banden gebruikt voor de reconstructie van lage frequenties kan dus veroorzaakt worden door de hoge gemiddelde energie op deze frequenties. Dit zou betekenen dat het NMF-algoritme vooral belang hecht aan de energie in plaats van informatie-inhoud op deze frequenties. Om dit na te gaan werd een weging toegepast op de rijen van $\mathbf{V}$, opdat de gemiddelde energie op elke frequentie over de gehele matrix gelijk wordt (dit betekent dat de grafiek in figuur 4.19 volledig vlak wordt). In dit geval wordt er een triviale factorisatie gevonden gelijkaardig aan figuur 4.3, met een heel geringe toename in breedte bij hogere frequenties. Dit lijkt erop te wijzen dat NMF energie meet in plaats van informatie.

Dit probleem moet echter genuanceerd worden. Het feit dat de banden breder worden naarmate hogere frequenties worden beschouwd, is vooral te wijten aan een kleinere informatie-inhoud in de overeenkomstige frequentie-gebieden. In het vervolg van deze sectie worden een aantal argumenten aangehaald die dit aantonen.

Gering verband tussen energieverloop en bandbreedtes

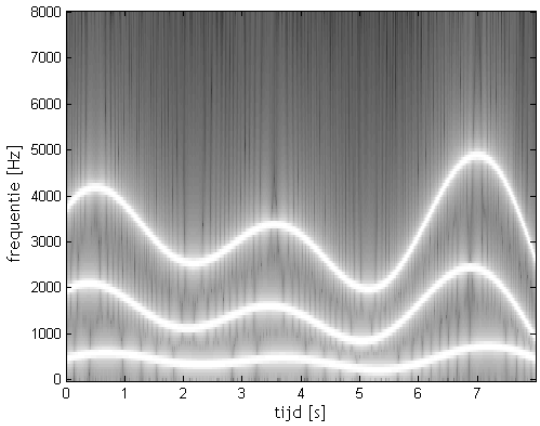
Op de eerste plaats is het nuttig om figuur 4.21 en figuur 4.19 te vergelijken. Er is geen duidelijk rechtstreeks verband tussen beide figuren. Dit is al een eerste aanwijzing dat de breedte van de banden niet volledig door de gemiddelde energie-inhoud verklaard kan worden. De hoog-energetische piek rond 800 Hz in figuur 4.19 komt niet tot uiting in de breedtes van de banden die door NMF worden gevonden. Bovendien blijkt uit figuur 4.21 dat de sterkste stijging in breedte vanaf 4500 Hz gebeurt. Dit komt veel minder tot uiting in figuur 4.19. Indien NMF enkel energie zou meten zou deze sterke stijging in figuur 4.21 vanaf 800 Hz moeten gebeuren, aangezien daar de afname in energie het sterkst is.

Energie en informatie

Een weging van de rijen van $\mathbf{V}$ om de gemiddelde energie per frequentie uniform te maken is in principe niet correct. Hoewel de begrippen energie en informatie niet dezelfde zijn, is er toch een verband. De gemiddelde energie op elke frequentie is namelijk afhankelijk van 2 aspecten:

1. De hoogte van de pieken die op de respectievelijke frequentie verschijnen.

[25] Merk op dat er een verschil is met het MSE-criterium. MSE heeft een kwadratische afhankelijkheid van de energie. Bijgevolg is er een volledige dominantie van een klein aantal hoge pieken. Het divergentie-criterium is slechts lineair afhankelijk van energie. Het divergentie-criterium behoudt daarom een globale visie op de volledige matrix, in plaats van een lokale focus op hoog-energetische pieken.



Figuur 4.22: Schematische voorstelling van de verplaatsing van de drie eerste formanten in een spraaksignaal

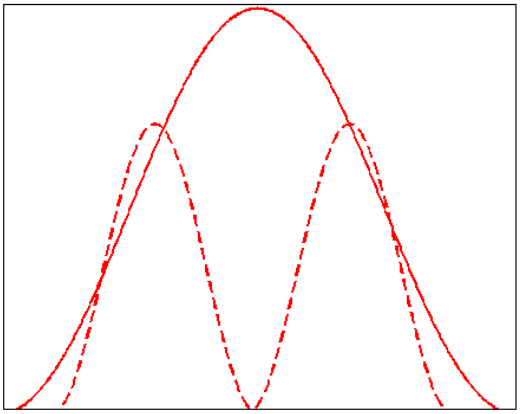
2. Het aantal keer dat er een piek op de respectievelijke frequentie verschijnt.

Het eerste aspect is slechts licht gerelateerd tot informatie-inhoud. Een hoog energetisch gebied is significanter en stijgt hoger boven het ruisniveau uit. Grote pieken hebben voor een waarnemer dus een grotere informatieve waarde dan kleine pieken. Merk op dat hoge pieken vaak kleinere pieken maskeren, zodat de informatie-inhoud van de kleine pieken verdwijnt.

Het tweede aspect is veel sterker gerelateerd aan informatie-inhoud. Indien op een bepaalde frequentie regelmatig een piek verschijnt, betekent dit dat er voor bepaalde klanken consistent een formant op deze frequentie staat, en zal de gemiddelde energie op deze frequentie ook hoger zijn. Voor meer ruizige klanken, zoals fricatieven, liggen de pieken niet consistent op dezelfde frequenties, maar liggen ze eerder willekeurig verspreid in het hoogfrequent gedeelte van het spectrum. Bijgevolg kan weinig informatie-inhoud aan de positie van dergelijke pieken toegekend worden en mogen deze pieken met brede banden gemodelleerd worden.

Het tweede aspect is ook van toepassing op het verschil tussen de variaties in positie van de opeenvolgende formanten in het spectrum. Ook hier is er een verschil in informatie-inhoud dat leidt tot een verschil in gemiddelde energie tussen de frequenties. Indien een typisch spraakspectrogram wordt geanalyseerd is het duidelijk dat de eerste formant over een veel kleiner frequentiegebied varieert dan de tweede en de derde formant. Dit is schematisch weergegeven in figuur 4.22. Deze figuur geeft een artificiële voorstelling van het voorloop van de eerste drie formanten. Aangezien de plaats van de eerste formant slechts licht varieert, zijn er smallere banden nodig om deze formant te volgen. Voor de tweede kunnen bredere banden gebruikt worden. De derde formant varieert over het algemeen nog veel sterker in frequentie. Het feit dat de gemiddelde energie op de lage frequenties groot is, is dus niet enkel omdat de pieken in het spectrum energetischer zijn, maar vooral omdat de pieken minder variëren in positie. Dit is in te zien aan de hand van figuur 4.22. De gemiddelde energie op 500 Hz zal duidelijk veel groter zijn dan op 3000 Hz, zelfs indien de pieken overal dezelfde hoogte hebben.

Dit alles toont aan dat informatie-inhoud en energie in feite verweven zijn. In teken van informatie-inhoud is het dus correct om smalle banden te plaatsen op de frequenties met een hoge gemiddelde energie. Indien de rijen van $\mathbf{V}$ gewogen worden opdat de gemiddelde energie op elke frequentie gelijk is, worden ook de verschillen in informatie-inhoud gecompenseerd. Door het toekennen van extra



Figuur 4.23: Een reconstructie van twee pieken met een breedbandige basisvector

energie aan de frequentiegebieden met een grote willekeur in de positie van de pieken, zal het NMF-algoritme ook meer aandacht besteden aan de reconstructie van het spectrum in deze frequentiegebieden. Dit is ten nadele van een goede modellering van de lage frequenties, waar de meeste structuur en determinisme aanwezig is.

4.6.3 Relatieve fout als criterium

Een relatieve fout als kostfunctie lijkt de optimale manier te zijn om alle bovenstaande problemen van energie-afhankelijkheid te omzeilen. Een relatieve fout maakt geen onderscheid tussen grote of kleine pieken. Het goede nieuws is dat, mits een aanpassing van de update-formules voor NMF met het divergentie-criterium, een kostfunctie kan geminimaliseerd worden die de relatieve fout benadert. In appendix A worden deze nieuwe update-formules wiskundig afgeleid. Het slechte nieuws is echter dat dit nieuwe algoritme het bovenstaande probleem niet kan oplossen. De kostfunctie is namelijk zodanig dat reconstructies steeds overschattingen van het te reconstrueren spectrum zijn[26]. Indien twee naburige pieken met slechts één brede band moeten gemodelleerd worden, worden grote relatieve fouten gemaakt op de tussenliggende waarden. Dit is weergegeven in figuur 4.23. De kleine waarden tussen de twee pieken zorgen ervoor dat de relatieve fout op deze plaatsen heel sterk wordt opgeblazen. Bijgevolg zal het NMF-algoritme ervoor zorgen dat er zo weinig mogelijk brede banden in de basisvectoren van de matrix $\mathbf{W}$ terecht komen. Bijgevolg wordt opnieuw de triviale oplossing zoals in figuur 4.3 bekomen.

4.6.4 Weging volgens gemiddelde hoogte van pieken

Een normalisatie die de gemiddelde hoogte van de pieken in de matrix op elke frequentie gelijk maakt, lijkt een goede oplossing te zijn. Dit zou de energie-verschillen tussen de pieken compenseren, zonder dat hierdoor de informatie-inhoud sterk wordt veranderd. Een dergelijke normalisatie is echter niet correct, aangezien klanken met veel pieken (ruizige klanken) op deze manier een sterk energetisch voordeel krijgen ten opzichte van klanken met slechts een drietal significante formanten (zoals klinkers). Hierdoor gaat de modellering van fricatieven (die heel weinig determinisme bevatten)

[26]Cfr. de redenering in sectie 4.5.1: een onderschatting (wat overeenkomt met een negatieve relatieve fout) zorgt steeds voor een veel grotere kost dan een overschatting, zoals figuur 4.10 aantoont.

domineren. Indien na de weging de energie per frame genormaliseerd wordt zoals beschreven in (4.1), wordt echter opnieuw een gelijkaardige factorisatie als de MEL-filterbank bekomen.

4.6.5 Normalisatie op basis van perceptie

Een energieweging die min of meer de structuur in de matrix bewaart, is een perceptuele normalisatie. De amplituderespons van het oor wordt vaak door een logaritmische compressie gemodelleerd[27]. Merk op dat dit een minder nauwkeurige benadering is dan de derde machtswortel die in sectie 4.5.3 als een voorverwerkingsmethode werd gebruikt.

Een normalisatie van de gemiddelde logaritmische energie voor alle frequenties kan dus beschouwd worden als het normaliseren van de data opdat het oor gelijke energie waarneemt voor alle frequenties. Dit komt neer op een weging van de rijen van $\mathbf{V}$ met de vector $\mathbf{w}$ waarbij

$$\mathbf{w}_i = \exp\left(\frac{1}{N}\sum_{j=1}^{N}\log v_{ij}\right) \tag{4.4}$$

met N het aantal kolommen van $\mathbf{V}$.

Na het doorvoeren van een dergelijke normalisatie is de verhouding tussen de frequentie met laagste gemiddelde energie en die met de hoogste gemiddelde energie slechts gelijk aan 2 (vergelijk dit met een factor van ongeveer 6 in figuur 4.19). Na een factorisatie met NMF worden opnieuw basisvectoren bekomen die op de MEL-filterbank gelijken, en weinig verschillen ten opzichte van de oorspronkelijke factorisatie.

4.6.6 Conclusie

Het achterhalen of NMF alleen maar energie meet, of toch rekening houdt met structuur in $\mathbf{V}$, blijkt een moeilijk probleem te zijn. De energie-inhoud en de informatie-inhoud in $\mathbf{V}$ zijn namelijk met elkaar verweven. Pogingen om de energie te normaliseren blijken ook de informatie-inhoud te veranderen. Een matrix-factorisatie op basis van relatieve reconstructiefouten blijkt ook niets op te leveren. Enkel bij een perceptuele normalisering van de energie blijkt dat de informatie-inhoud bewaard blijft.

Het lijkt er echter op dat de energie in de oorspronkelijke matrix $\mathbf{V}$ op een goede manier verdeeld is, opdat het divergentie-criterium een juiste structuur kan vinden. Een observatie van de spectra in de matrix $\mathbf{V}$ geeft namelijk aan dat het verschil tussen de hoogte van pieken op lage en hoge frequenties, het energieverloop in figuur 4.19 niet kan verklaren. Het energieverloop in deze figuur blijkt vooral een gevolg te zijn van de structuur van de data: de verschillende frequentievariaties van de opeenvolgende formanten en de willekeurige plaats van de pieken bij ruizige klanken. Een weging van de spectra op basis van deze figuur is dan ook incorrect en misleidt het NMF-algoritme. Dit leidt tot een triviale factorisatie die geen structuur verklaart in de data en ook een slechtere PER oplevert.

Er kan dus besloten worden dat de matrix-factorisatie op basis van informatie-inhoud gebeurt, ondanks de niet-uniforme energieverdeling over de frequenties. Bijgevolg kan gesteld worden dat het resultaat van dit experiment de stelling ondersteunt die in het begin van deze sectie werd geponeerd: productie

[27] Denk aan de deciBel schaal.

en analyse van spraak zijn goed op elkaar afgestemd om de informatie-overdracht zo efficiënt mogelijk te maken.

4.7 Besluit

Het doel van de experimenten in dit hoofdstuk was het vinden van een kenmerkenset op een automatische manier in de hoop dat deze betere herkenningsresultaten oplevert dan de MEL-filterbank die vaak wordt gebruikt voor kenmerken-extractie in spraakherkenningssystemen.

Via SWO werd een DCT gevonden, wat een vaak gebruike kenmerken-extractor is in spraakherkenningssystemen. NMF met MSE-criterium vond een moeilijk interpreteerbare kenmerkenset wegens een voorafgaande log-compressie op de spraakspectra.

Via NMF met divergentie-criterium werd een kenmerkenset gevonden die even goede PER resultaten oplevert als bij het gebruik van de MEL-filterbank. De gevonden kenmerkenset gelijkt bovendien sterk op de MEL-filterbank. Dit betekent dat de MEL-filterbank, die na vele jaren onderzoek binnen spraakverwerking een van de meest praktische kenmerken-extractors blijkt te zijn, ook gevonden wordt via een zelflerend algoritme op basis van observaties van spraaksignalen.

Aangezien een gelijkaardig resultaat wordt bekomen door analyse van de frequentieresolutie van het oor enerzijds (MEL schaal) en analyse van spraaksignalen anderzijds, toont dit resultaat aan dat spraak goed is afgestemd op het menselijk gehoorsysteem. Dit is niet verwonderlijk: het menselijke spraaksysteem heeft immers een lange evolutie ondergaan en is uitgegroeid tot een heel efficiënt communicatiemiddel. Men kan daarom de hypothese poneren dat de productie van menselijke spraak geëvolueerd is opdat de informatie, die nodig is om spraak te herkennen, zo goed mogelijk behouden blijft na analyse door het gehoorsysteem. We produceren klanken op een zodanige manier dat het oor deze informatie zo goed mogelijk kan extraheren.

Bovendien bevestigen de resultaten de hypothese van Lee & Seung [6] dat de patronen die door een niet-negatieve matrix-factorisatie blootgelegd worden vaak goed overeenstemmen met menselijke perceptie van de te interpreteren data.

Hoofdstuk 5

Zelflerende foneemclassificatie

5.1 Inleiding

Om spraakherkenningssystemen te trainen is een grote set met spraaksignalen nodig, waarbij elk foneem een voldoende aantal keer in de trainingset moet voorkomen. Bovendien is er nood aan een foneemtranscriptie van de data[1]. Deze transcriptie geeft voor elk tijdstip aan welk foneem wordt uitgesproken. Op basis van deze informatie kan voor elk foneem een HMM opgesteld worden door een spraakherkenningssysteem.

Wanneer de mens in zijn eerste levensjaren spraak leert begrijpen, wordt er uiteraard geen gebruik gemaakt van een dergelijke foneemtranscriptie. Het brein moet zonder foneemkennis een goede classificatie vinden voor de akoestische eenheden in spraaksignalen. In dit hoofdstuk wordt nagegaan of een computer ook in staat is om spraaksegmenten, die afkomstig zijn van hetzelfde foneem, te groeperen indien geen a-priorische foneemkennis mag gebruikt worden.

Om de probleemstelling op te lossen wordt verondersteld dat kenmerkenvectoren van frames, die afkomstig zijn van dezelfde akoestische eenheid, dicht bij elkaar liggen in de kenmerken-ruimte. Op het eerste zicht lijkt clustering een ideale methode om dit probleem op te lossen. Clusteringsalgoritmes zijn echter sterk afhankelijk van de afstandsmaat die gebruikt wordt. De experimenten in dit hoofdstuk zullen zo opgesteld worden dat deze afhankelijkheid van de onderliggende afstandsmaat te verwaarlozen is. Dit gebeurt aan de hand van een KNN-matrix ('k *nearest neighbours*'). Deze matrix geeft voor elk punt de k punten die het dichtst bij het respectievelijke punt gelegen zijn in de kenmerken-ruimte. Op deze manier wordt geclusterd op basis van buren, in plaats van een clustering op basis van afstand.

Er worden twee verschillende oplossingsmethoden uitgeprobeerd. De eerste is gebaseerd op de matrix-factorisatie algoritmes die in het vorige hoofdstuk werden toegepast. De tweede methode is gebaseerd op een eigenwaardenprobleem met niet-negativiteitsvoorwaarden. Beide methoden hebben een extra voordeel ten opzichte van clustering: het onderliggende model is robuust tegen klanken die door co-articulatie niet aan één bepaald foneem toegewezen kunnen worden. Dergelijke klanken hebben een bijdrage van meerdere fonemen. Clusteringsalgoritmes zullen door dergelijke klanken in verwarring gebracht worden omdat elke klank tot slechts één cluster kan behoren. Matrix-factorisatietechnieken

[1] Deze transcriptie wordt meestal op een automatische manier opgesteld en door experts geverifieerd.

daarentegen hebben de vrijheid om klanken te verklaren door middel van meerdere basisvectoren.

5.2 Opstellen van de KNN-matrix V

5.2.1 Definitie van de KNN-matrix

De $(N \times N)$ KNN-matrix $\mathbf{V}$ voor de verzameling kenmerkenvectoren $P = \{P_1, ..., P_N\}$ wordt gedefinieerd als:

$$\mathbf{V}_{ij} = \begin{cases} \frac{D_j}{d(i,j)^\alpha} & \text{als} \quad P_i \in \text{KNN}(P_j) \\ 0 & \text{als} \quad P_i \notin \text{KNN}(P_j) \end{cases} \qquad i = 1...N,\ j = 1...N \tag{5.1}$$

Hierbij is $d(i, j)$ de Euclidische afstand tussen de punten P_i en P_j. KNN(P_j) is de verzameling van de k punten die het dichtst bij punt P_j gelegen zijn volgens een Euclidische afstandsmaat. α is een tuning-parameter. D_j is de gemiddelde afstand tot de l punten die het dichts bij het punt P_j gelegen zijn.

De constructie van de matrix $\mathbf{V}$ zorgt ervoor dat de l punten die het dichtst bij punt P_j liggen een waarde van ongeveer 1 krijgen. De waarde van de k dichtste punten daalt volgens een macht van hun inverse afstand tot punt P_j. De sterkte van de afname kan geregeld worden met parameter α.

5.2.2 Aanpassing voor spraakdata

De TIMIT-databank is niet gebalanceerd qua foneeminhoud aangezien dit een databank is met continue spraak. Daarom wordt $\mathbf{V}$ anders gedefinieerd, waarbij de betekenis van k verandert:

$$\mathbf{V}_{ij} = \begin{cases} \frac{D_j}{d(i,j)^\alpha} & \text{als} \quad \frac{D_j}{d(i,j)} \geq v_{clip} \\ 0 & \text{als} \quad \frac{D_j}{d(i,j)} \leq v_{clip} \end{cases} \qquad i = 1...N,\ j = 1...N \tag{5.2}$$

waarbij v_{clip} een afkappingsconstante is. Deze constante wordt zo bepaald dat er gemiddeld k elementen in elke kolom van $\mathbf{V}$ staan. Het aantal niet-nul elementen is dus verschillend voor elke kolom van $\mathbf{V}$. Hierdoor zullen punten die afkomstig zijn van een foneem met veel voorbeelden in de databank ook veel niet-nul elementen hebben in hun respectievelijke kolom. Bovendien wordt op deze manier gegarandeerd dat er enkel rekening wordt gehouden met buren die effectief ook dicht bij het beschouwde punt liggen. In een KNN-matrix volgens (5.1) kunnen punten, die bij een foneem horen met slechts heel weinig voorbeelden, een groot aantal buren hebben die niet tot dezelfde foneemklasse behoren.

Tot slot wordt steeds de normalisatie (4.1) uitgevoerd op de kolommen van $\mathbf{V}$ opdat het matrix-factorisatie algoritme aan de reconstructie van alle kolommen evenveel belang hecht.

5.2.3 Parameterkeuze

Omwille van geheugenproblemen wordt de TIMIT databank opgesplitst in mannelijke en vrouwelijke sprekers. De berekeningen gebeuren op de vrouwelijke sprekers. In dat geval geldt dat $N \approx 300000$.

De keuze van k is cruciaal in deze experimenten. Indien de waarde van k te klein is, zijn er te weinig gemeenschappelijke elementen in de verzamelingen met k dichtste buren van de verschillende punten binnen één klasse. In dat geval is het voor de methoden die hierna worden besproken moeilijk om structuur te ontdekken. Indien de waarde van k echter te groot wordt gekozen, zullen veel van de k dichtste buren niet tot dezelfde foneemklasse behoren, wat ongewenst is. De waarde van k wordt bovendien beperkt door het RAM-geheugen van de computer die met de matrix $\mathbf{V}$ moet rekenen. Dit laatste blijkt de belangrijkste beperking te zijn bij de experimenten die in dit hoofdstuk worden besproken. In de experimenten waarvan de resultaten in de tekst zijn opgenomen, werd steeds $k = 256$ gekozen. Hierbij werd rekening gehouden met bovenstaande opmerkingen. Deze waarde vormt een goed evenwicht tussen performantie en rekentijd. De kwaliteit van de classificatie van de spraaksegmenten op basis van een KNN-matrix met $k > 256$ is niet beter[2] dan op basis van een KNN-matrix met $k = 256$.

In de experimenten op de spraakdata wordt l steeds gelijk gesteld aan $\sqrt{k}$. Dit is een pragmatische keuze, en blijkt een goede waarde te zijn om een adequate schatting te bekomen van de spreiding van de punten in het gebied rond het beschouwde punt. Voor parameter α wordt de waarde 2 gekozen. Dit zorgt ervoor dat er een redelijk snelle afname is van de k elementen in de respectievelijke kolom. Toch blijkt dat deze afname in de meeste kolommen niet snel genoeg is om 0 te bereiken. Indien de parameter α echter groter wordt gekozen, zal de afname te snel zijn. In dat geval zijn de waarden van de k dichtste buren in de matrix te klein om nog een structuur terug te vinden met de twee methodes die verder in dit hoofdstuk worden besproken. Om toch min of meer een afname tot nul te bereiken wordt daarom een constante afgetrokken van de niet-nul elementen in de matrix. Om de niet-negativiteit van de matrix te behouden worden de negatieve elementen gelijk gesteld aan nul.

5.3 Foneemclassificatie via niet-negatieve matrix-factorisatie

5.3.1 Matrix-factorisatie op basis van het PLSA-model

In deze sectie is het de bedoeling om een niet-negatieve matrix-factorisatie uit te voeren van de KNN-matrix $\mathbf{V}$. Het probleem van foneemclassificatie is namelijk analoog als de probleemstelling van LSA, die in hoofdstuk 3 werd besproken. Hierbij werd de '*word-counts*' matrix (3.3) gefactoriseerd. Om negatieve gewichten te vermijden werd het PLSA-model geïntroduceerd. Het PLSA-model kan semantische relaties blootleggen tussen documenten, zelfs als deze geen enkel woord gemeenschappelijk hebben [7, 8]. Dit gebeurt via overlap met andere documenten, die een connectie vormen tussen twee niet-overlappende documenten.

Dit laatste vormt de onderliggende basis voor dit experiment. Punten die geen k dichtste buren gemeenschappelijk hebben, maar wel tot dezelfde foneemklasse behoren, kunnen volgens het PLSA-model toch door hetzelfde aspect of dezelfde basisvector verklaard worden.

De $(N \times N)$ matrix $\mathbf{V}$ wordt benaderd door het product van de $(N \times r)$ matrix $\mathbf{W}$ en de $(r \times N)$ matrix $\mathbf{H}$, waarbij het divergentie-criterium als kostfunctie gebruikt wordt. Hierbij wordt $r = 51$ gekozen, in de hoop dat de gevonden aspecten overeenkomen met de foonklassen die in de transcriptie van de TIMIT databank gebruikt worden. De foonidentiteit van een spraaksegment is namelijk de verborgen

[2] Om dit na te gaan werd een experiment met $k = 512$ uitgevoerd. De kwaliteit werd bepaald op basis van perplexiteit (zie verder).

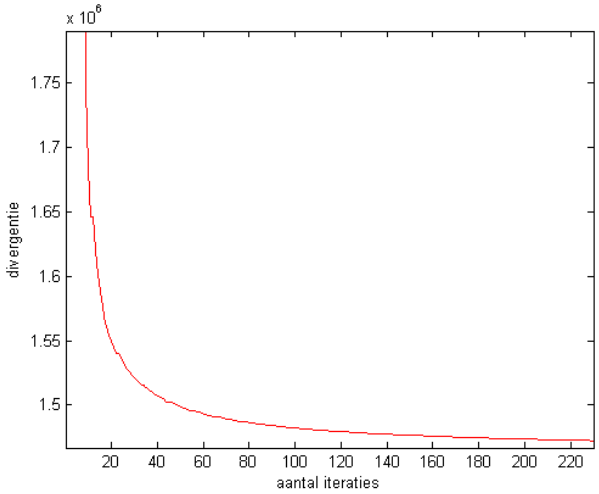
variabele die de klank, en bijgevolg de plaats in de kenmerken-ruimte, bepaalt. De gewichten in de kolommen van matrix $\mathbf{H}$ geven dan aan in hoeverre het beschouwde frame tot een bepaalde foonklasse behoort.

Merk op dat het hier gaat om fonen in plaats van om fonemen. In de Engelse taal zijn er slechts 43 fonemen. De manuele transcriptie van de spraaksignalen in de TIMIT databank bevat echter 51 labels. Zo wordt bv. een onderscheid gemaakt tussen stemhebbende en stemloze '*closures*' van plosieven. Ook stilte-frames krijgen een label. In bijlage B wordt de volledige lijst opgesomd van de labels die in de transcriptie van de TIMIT databank gebruikt worden. In het vervolg van de tekst wordt dan ook over foonklassen gesproken in plaats van foneemklassen.

5.3.2 Rekentijd en aantal iteraties

De KNN-matrix $\mathbf{V}$ bevat veel meer elementen dan de matrix $\mathbf{V}$ uit het vorige hoofdstuk. De meeste elementen zijn echter gelijk aan nul, zodat de matrix compact kan beschreven worden via een 'spaarse' representatie (*sparse representation*). Het NMF-algoritme werd aangepast om met een dergelijke representatie overweg te kunnen. Bepaalde berekeningen moeten hierbij in C gebeuren, omdat MATLAB deze niet efficiënt kan uitvoeren.

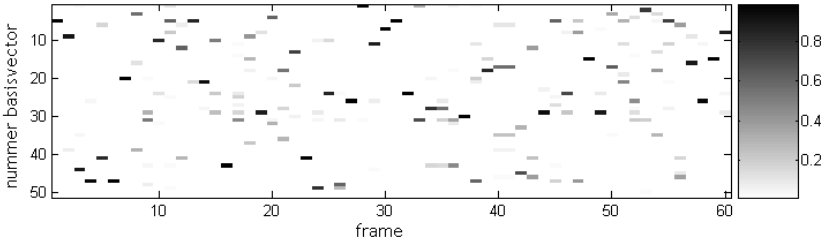
De rekentijd bedraagt ongeveer 400 seconden per iteratie op een 2,4GHz dual AMD opteron 280 processor met 16GB DDR400 geheugen. Dit is 8 keer langer dan de rekentijd van de experimenten uit het vorige hoofdstuk. Om de totale rekentijd te beperken werden telkens slechts een 200-tal iteraties uitgevoerd. De kostfunctie is dan nog niet helemaal geconvergeerd, maar reeds voldoende om een beeld te hebben van de uiteindelijke geconvergeerde factorisatie. Het verloop van de kostfunctie in functie van het aantal iteraties wordt weergegeven in figuur 5.1.



Figuur 5.1: Divergentie in functie van het aantal uitgevoerde iteraties bij het KNN-experiment

5.3.3 Resultaten

Uit de analyse van de kolommen van $\mathbf{H}$ blijkt dat elke kolom slechts één à drie dominante waarden bevat waarvan één meestal meer uitgesproken is (zie figuur 5.2). Dit wijst erop dat de meeste punten vooral door één klasse worden verklaard. Dit is wat verwacht werd: de meeste frames zijn een uiting van één bepaald foon. Door co-articulatie kunnen bepaalde frames ook door meerdere foonklassen verklaard worden.



Figuur 5.2: 60 willekeurige kolommen van de matrix $\mathbf{H}$. De kolommen werden genormaliseerd zodat de som in elke kolom gelijk is aan 1.

Kwaliteit van het model op basis van perplexiteit

Om de kwaliteit van het model te evalueren wordt een vergelijking gemaakt met de manuele[3] transcriptie van de spraaksignalen in de TIMIT databank. Deze transcriptie associeert elk frame met een bepaald foon. Op basis hiervan wordt een matrix $\mathbf{F}^{\mathrm{s}}$ gedefinieerd:

$$\mathbf{F}^{\mathrm{s}}_{ij} = \sum_{m \in \Phi(i)} \mathbf{H}_{jm} \tag{5.3}$$

waarbij $\Phi(i)$ de verzameling is met alle indices van de kolommen van $\mathbf{H}$ die bij foon i behoren volgens de manuele foontranscriptie van de TIMIT-databank. De kolommen van deze matrix komen overeen met de indices van de r basisvectoren in de kolommen van $\mathbf{W}$. De 51 rijen van $\mathbf{F}^{\mathrm{s}}$ komen overeen met foonlabels. Aangezien $r = 51$ is $\mathbf{F}^{\mathrm{s}}$ een vierkante matrix. Het superscript 's' staat voor '*soft*', omdat deze matrix informatie verzamelt over zachte classificatiebeslissingen. Er wordt bovendien een variant van deze matrix gedefinieerd: $\mathbf{F}^{\mathrm{h}}$. Element $\mathbf{F}^{\mathrm{h}}_{ij}$ van deze matrix is gelijk aan het aantal keer dat een kolommaximum wordt bereikt op rij-index j in de kolommen van $\mathbf{H}$ met kolom-index $m \in \Phi(i)$. Het superscript 'h' staat voor '*hard*', omdat deze matrix informatie over harde beslissingen verzamelt.

In het ideale geval zal de matrix $\mathbf{F}^{\mathrm{h}}$ in elke rij slechts één element met een waarde verschillend van nul hebben. Dit zou betekenen dat alle frames die afkomstig zijn van dezelfde foon ook dezelfde basisvector uit $\mathbf{W}$ gebruiken. Bovendien zal in het ideale geval ook in elke kolom van $\mathbf{F}^{\mathrm{h}}$ slechts één element staan met een waarde verschillend van nul. Dit betekent dat elke basisvector slechts met één foonklasse overeenkomt. De kwaliteit van het model is dus meetbaar via de perplexiteit van de

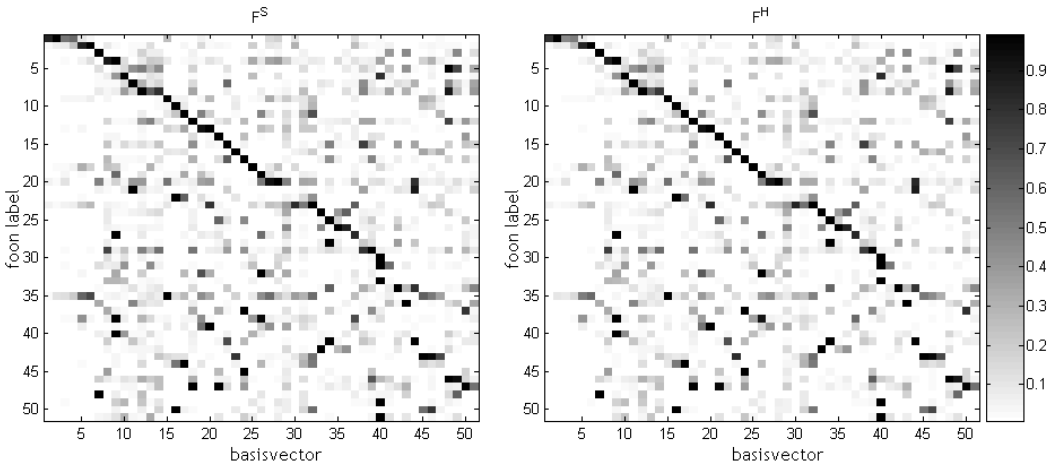
[3]Deze transcriptie werd op een automatische manier opgesteld en door experts geverifieerd. In het vervolg van de tekst wordt echter de term 'manuele' transcriptie behouden, om een onderscheid te maken met de automatische correctie van deze transcriptie op basis van de ESAT-spraakherkenner (zie sectie 5.3.5).

rijen en de kolommen van $\mathbf{F}^{\mathrm{h}}$. Hoe dichter al deze perplexiteiten bij 1 liggen, hoe beter[4]. Indien het model een perfecte classificatie uitvoert zijn al deze perplexiteiten gelijk aan 1, en is de matrix $\mathbf{F}^{\mathrm{h}}$ perfect diagonaliseerbaar[5]. De 'diagonaliseerbaarheid' van deze matrix is dus een goede maat voor de kwaliteit van het model.

Daar de manuele foonclassificatie van de TIMIT databank op basis van harde beslissingen gebeurt is het moeilijk een maat te definiëren die ook rekening houdt met de mogelijkheden van het model om co-articulatie te modelleren. Rekening houdend met het onderliggende PLSA-model kan er echter aangenomen worden dat het model geen problemen heeft met zachte beslissingen indien de harde beslissingen voldoende nauwkeurig zijn.

Figuur 5.3 toont de matrices $\mathbf{F}^{\mathrm{s}}$ en $\mathbf{F}^{\mathrm{h}}$ die werden opgesteld op basis van de matrix $\mathbf{H}$ die door NMF werd berekend. De rijen werden genormaliseerd zodat hun maximum de waarde 1 aanneemt. Daarna werden de kolommen gepermuteerd om de matrices zo goed als mogelijk te diagonaliseren. In appendix B staat de lijst die aangeeft welke rij-index met welk foon overeenkomt.

Het is duidelijk dat $\mathbf{F}^{\mathrm{s}} \approx \mathbf{F}^{\mathrm{h}}$. Dit wijst erop dat de kolommen van $\mathbf{H}$ meestal een dominante component hebben, die aangeeft tot welke foonklasse het frame behoort. De matrices kunnen echter moeilijk gediagonaliseerd worden. Bijna alle foonklassen gebruiken namelijk meer dan één basisvector. Dit betekent dat niet alle uitingen van hetzelfde foon door dezelfde basisvector verklaard worden. Bovendien wordt bijna elke basisvector door meerdere foonklassen gebruikt. Dit wijst erop dat het model er niet volledig in slaagt om de verschillende foonklassen te scheiden.



Figuur 5.3: De gediagonaliseerde matrices $\mathbf{F}^{\mathrm{s}}$ (links) en $\mathbf{F}^{\mathrm{h}}$ (rechts) na matrix-factorisatie van de KNN-matrix $\mathbf{V}$ met NMF volgens het divergentiecriterium.

[4]Merk op dat zowel rijen als kolommen een lage perplexiteit moeten hebben. Indien enkel de rijen een heel lage perplexiteit hebben, betekent dit niet dat een goed model werd gevonden. Dit is bijvoorbeeld het geval als alle foonklassen voor alle spraakframes dezelfde basisvector gebruiken.

[5]Dit alles is in de veronderstelling dat de manuele transcriptie van de TIMIT-databank perfect is. Dit zal nooit het geval zijn omdat er steeds spraakframes tussen 2 fonen liggen. Het is dan vaak moeilijk om objectief te beslissen aan welk foon dergelijke spraakframes worden toegewezen.

Kwaliteit van het model op basis van een foneemherkenningsexperiment

Naast de 'diagonaliseerbaarheid' van $\mathbf{F}^{\text{h}}$ kan ook de voorspellende kracht als maat dienen voor de kwaliteit van het model. Hiervoor wordt een *multi-layer-perceptron* (MLP) met 2 lagen gebruikt. Dit is een artificieel neuraal netwerk dat als classificator dient.

Alvorens een classificatie kan uitgevoerd worden moet het MLP getraind worden. Dit gebeurt met de trainingset van de TIMIT databank (dit is ook de dataset waarvan de KNN-matrix reeds werd gefactoriseerd). Vervolgens kan een classificatie uitgevoerd worden op de trainingset (kolommen van $\mathbf{H}_{\text{train}}$) en op de testset (kolommen van $\mathbf{H}_{\text{test}}$). Deze matrices worden gedefinieerd als:

$$\mathbf{H}_{\text{test}} = \mathbf{W}^T\mathbf{V}_{\text{test}} \ , \qquad \mathbf{H}_{\text{train}} = \mathbf{W}^T\mathbf{V}_{\text{train}} \tag{5.4}$$

De matrix $\mathbf{W}$ is gevonden via het NMF-algoritme op de KNN-matrix $\mathbf{V}_{\text{train}}$ van de trainingset. De matrix $\mathbf{V}_{\text{test}}$ is de $(N_{train} \times N_{test})$ KNN-matrix van de testset, waarbij elke kolom de k dichtste buren uit de trainingset bevat, horende bij het beschouwde punt uit de testset. Het product van $\mathbf{W}^T$ met de KNN-matrix is analoog met het gebruik van de matrix $\mathbf{W}$ als filterbank zoals in het vorige hoofdstuk. Dit blijkt in dit geval ook betere resultaten te geven dan het gebruik van de matrix $\mathbf{H}$ die via een factorisatie wordt bekomen.

Eerst moet het aantal correcte classificatiebeslissingen bepaald worden indien de 'ideale' matrix $\mathbf{W}^I$ gebruikt wordt voor het bepalen van $\mathbf{H}_{\text{train}}$ en $\mathbf{H}_{\text{test}}$. Deze matrix wordt gedefinieerd als

$$\mathbf{W}^I_{ji} = \begin{cases} 1 & \text{als} \quad j \in \Phi(i) \\ \\ 0 & \text{als} \quad j \notin \Phi(i) \end{cases} \tag{5.5}$$

Deze matrix geeft 64,8% correcte classificaties van de kolommen van $\mathbf{H}_{\text{train}}$ en 61,8% voor $\mathbf{H}_{\text{test}}$. Deze twee waarden kunnen beschouwd worden als het streefdoel. Indien de matrixfactorisatie van $\mathbf{V}_{\text{train}}$ en $\mathbf{V}_{\text{test}}$ erin slaagt om foonklassen te vinden, moeten even goede classificatieresultaten behaald worden. In tabel 5.1 worden de classificatieresultaten weergegeven voor twee NMF-factorisaties van $\mathbf{V}_{\text{train}}$. Een daarvan werd geïnitialiseerd met een willekeurige matrix $\mathbf{W}$, de andere werd geïnitialiseerd met $\mathbf{W}^I$.

Tabel 5.1: Vergelijking tussen classificatieresultaten op basis van $\mathbf{W}^I$ en op basis van de matrix $\mathbf{W}$ die door NMF werd berekend met respectievelijk willekeurige initialisatie en met initialisatie $\mathbf{W}_0 = \mathbf{W}^I$.

	$\mathbf{W}^I$	$\mathbf{W}_{\text{NMF}}$	$\mathbf{W}_{\text{NMF}}$ met $\mathbf{W}_0 = \mathbf{W}^I$
trainset	64,8%	58.7%	59,6%
testset	61,8%	57.9%	58,7%

De laatste kolom van tabel 5.1 wijst erop dat de ideale oplossing niet stabiel is bij het toepassen van de NMF-updateformules. Een matrix-factorisatie met NMF zal dus de perfecte classificatie niet als oplossing kunnen vinden voor deze dataset.

5.3.4 Problemen met het PLSA-model

Uit de resultaten blijkt dat NMF of het PLSA-model geen goede methode is om tot een correcte oplossing van het zelflerend classificatieprobleem te komen. Dit is niet verwonderlijk, aangezien

NMF niet streeft naar een diagonale matrix $\mathbf{F}^{\text{h}}$. NMF zoekt namelijk punten in de KNN-ruimte waarmee een laagdimensionale simplex wordt opgespannen waarmee de kolommen van de KNN-matrix $\mathbf{V}$ kunnen gereconstrueerd worden[6]. De punten die hiervoor door NMF worden gekozen zijn heel sterk afhankelijk van de initialisatie. Punten die in de KNN-ruimte tussen meerdere foonklassen liggen kunnen ook dienen als hoekpunten van deze simplex en zijn ook in staat om een factorisatie te genereren met een lage divergentie tussen $\mathbf{V}$ en het product $\mathbf{WH}$. Dit is echter ongewenst, aangezien dit betekent dat de foonklassen waartussen het beschouwde hoekpunt zich bevindt, hetzelfde punt gebruiken om een gedeelte van hun data te verklaren. De overeenkomstige kolom van $\mathbf{W}$ zal dus dominante elementen bevatten afkomstig van verschillende foonklassen. Hierdoor is het niet mogelijk om $\mathbf{F}^{\text{h}}$ te diagonaliseren.

In figuur C-1 (zie kleurenbijlage) wordt dit geïllustreerd aan de hand van een artificieel clusteringsprobleem in een 2-dimensionale ruimte. De KNN-matrix, behorende bij de punten in deze artificiële dataset, werd via NMF gefactoriseerd. De kleur van elk punt in deze figuur wordt bepaald door de rij-index van het dominante gewicht in de overeenkomstige kolom van $\mathbf{H}$. Het blijkt dat de clusters door NMF op een arbitraire manier worden onderverdeeld, afhankelijk van de initialisatie. Dit wijst erop dat de hoekpunten van de simplex tussen verschillende klassen gelegen zijn in de KNN-ruimte.

Er is echter een tweede probleem met het gebruik van NMF of het PLSA-model: de perfecte classificatie is in teken van het divergentie-criterium suboptimaal, omdat er een groot verschil is in de grootte van de verschillende foonklassen. Het divergentie-criterium streeft namelijk naar een gelijk aantal dominante elementen in elke kolom van $\mathbf{W}$. De reden hiervoor is dat in elke kolom van $\mathbf{V}$ slechts een klein aantal elementen verschillend zijn van nul. Voor de eenvoud wordt even verondersteld dat $\mathbf{V}$ per kolom k elementen heeft met waarde 1, en voor de rest enkel uit nullen bestaat. Stel dat kolom $\mathbf{w}_i$ van $\mathbf{W}$ N_i dominante elementen bevat. Dan zullen ongeveer N_i kolommen van $\mathbf{H}$ een groot gewicht toekennen aan $\mathbf{w}_i$ bij de reconstructie van $\mathbf{V}$. Deze N_i kolommen zullen door kolom $\mathbf{w}_i$ gereconstrueerd worden met $N_i - k$ fouten per kolom. Bijgevolg is het totale aantal significante fouten bij een perfecte classificatie gelijk aan $\sum_i N_i \left(N_i - k\right)$. Dit aantal is minimaal als N_i gelijk is aan k voor alle i. Op praktische data zal het NMF-algoritme een lokaal minimum vinden, waarbij $N_i >> k$. Ook hier geldt echter dat een kleinere waarde voor het divergentie-criterium bereikt wordt als N_i ongeveer gelijk is voor alle i. Bijgevolg zal het divergentie-criterium bij een willekeurige initialisatie niet de tendens hebben om de kolommen van $\mathbf{W}$ op een onevenwichtige manier te verdelen met dominante elementen, zoals dat bij een perfecte classificatie het geval is. Dit blijkt zowel uit experimenten met spraakdata, als uit artificiële experimenten zoals in figuur C-1. De winst door overlap van de dichtste buren weegt niet op tegen de tendens om in elke kolom van $\mathbf{W}$ evenveel dominante elementen te plaatsen. De correcte classificatie kan bijgevolg moeilijk gevonden worden door NMF in het geval van een willekeurige initialisatie.

Deze beide problemen zijn echter niet de oorzaak van het feit dat de ideale oplossing niet stabiel is bij het toepassen van de NMF-updateformules, als de KNN-matrix met spraakdata gebruikt wordt. Uit de experimenten met de artificiële dataset uit figuur C-1 blijkt dat de correcte oplossing stabiel is onder de NMF updateformules, ondanks het verschil in de grootte tussen de vier klassen. Dit betekent dat de correcte oplossing in de buurt ligt van een lokaal minimum van de kostfunctie. Het feit dat dit bij spraakdata niet geldt, heeft dus te maken met de dataset zelf. Er is vermoedelijk een grote overlap in de kenmerkenruimte tussen bepaalde foonklassen, waardoor de perfecte foonclassificatie geen stabiele oplossing is van een factorisatie van de KNN-matrix.

[6]De hoekpunten van deze simplex worden bepaald door de kolommen van de matrix $\mathbf{W}$.

5.3.5 Interpretatie van de resultaten

Als conclusie kan gesteld worden dat er 3 mogelijke oorzaken meespelen voor het feit dat er geen perfecte classificatie werd gevonden:

1. De manuele foontranscriptie die bij de TIMIT databank wordt meegegeven is niet foutloos. Aangezien de kwaliteit van het model aan de hand van deze informatie wordt bepaald, kan het zijn dat bepaalde frames, die door het model goed gemodelleerd worden, toch als fout worden geklasseerd volgens de manuele foontranscriptie.
2. Het PLSA-model heeft enkele gebreken waardoor moeilijk een classificatie van hoge kwaliteit bereikt kan worden.
3. Bepaalde foonklassen hebben een te grote overlap in de kenmerken-ruimte waardoor het vrijwel onmogelijk is om deze overlappende gebieden te scheiden via een KNN-model.

Het is moeilijk om te achterhalen hoe sterk de eerste oorzaak meespeelt. Er werd een nieuwe matrix $\mathbf{F}^{\mathrm{h}}$ opgesteld op basis van dezelfde foontranscriptie, waarbij de foongrenzen gecorrigeerd werden door een automatisch systeem dat betrouwbaarder is dan de manuele transcriptie[7]. Deze matrix is quasi identiek aan de matrix $\mathbf{F}^{\mathrm{h}}$ op basis van de manuele transcriptie die bij de TIMIT databank geleverd wordt. Het feit dat beide transcripties een gelijkaardig resultaat geven wijst er echter op dat de kwaliteit van de transcriptie niet de belangrijkste oorzaak is van de hoge perplexiteiten in de rijen en kolommen van $\mathbf{F}^{\mathrm{h}}$.

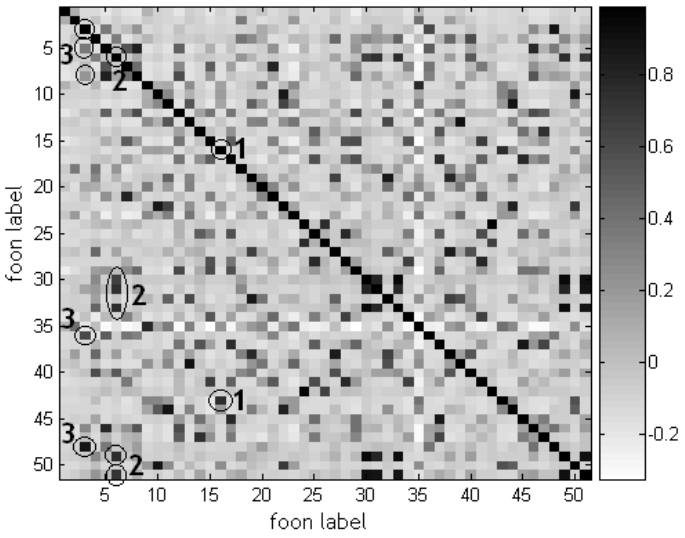
Omtrent de tweede oorzaak is er weinig twijfel. Zoals in sectie 5.3.4 werd aangetoond heeft PLSA een aantal niet te vermijden problemen waardoor moeilijk een perfecte classificatie kan bereikt worden.

Om de impact van de derde oorzaak te achterhalen is het interessant om de correlatie-coëfficiënten te analyseren tussen de rijen van de matrix $\mathbf{F}^{\mathrm{h}}$. Deze zullen hoog zijn indien de twee beschouwde foonklassen vaak door dezelfde basisvectoren verklaard worden. Figuur 5.4 toont de matrix met deze correlatie-coëfficiënten. Het blijkt dat de hoge correlatiewaarden in de matrix vaak verklaard kunnen worden door de gelijkenis in klank en uitspraak van de fonen die beschouwd worden. Ter illustratie werden drie dergelijke relaties in de figuur aangeduid:

1. Foonlabels 16 en 43 overeenkomstig de respectievelijke fonen [s] en [z].
2. Foonlabels 6, 30, 31, 33, 49 en 51 overeenkomstig de respectievelijke fonen [n], [,m], [N], [m], [,n] en [,N].
3. Foonlabels 3, 5, 8, 36 en 48 overeenkomstig de respectievelijke fonen [i], [E], [I], [E+j] en [j].

Vermoedelijk betekenen de hoge correlatiecoëfficiënten tussen bepaalde foonklassen dat deze foonklassen wegens de gelijkenissen in klank door elkaar liggen in de kenmerken-ruimte. In dat geval kan het KNN-model vrijwel onmogelijk tot een perfecte classificatie leiden. Zoals in sectie 5.3.4 werd uitgelegd, is het feit dat de correcte classificatie geen stabiel punt is van NMF ook een aanwijzing dat er veel overlap is tussen bepaalde foonklassen.

[7] Dit is op basis van de ESAT-spraakherkenner [19].



Figuur 5.4: Correlatiecoëfficiënten tussen de rijen van de matrix $\mathbf{F}^{\text{h}}$ na het toepassen van NMF op de KNN-matrix

5.4 Foneemclassificatie op basis van eigenwaardenontbinding

5.4.1 Klassieke eigenwaardenontbinding van de KNN-matrix

In deze sectie wordt een andere methode voorgesteld om via de KNN-matrix $\mathbf{V}$ tot een foneem- of foonclassificatie te komen. Deze methode is geïnspireerd op een eigenwaardenontbinding van $\mathbf{V}^T$. In de context van KNN is een eigenwaardenontbinding namelijk een interessante methode voor het opstellen van de matrix $\mathbf{H}$. Indien de rijen van deze matrix eigenvectoren zijn van $\mathbf{V}^T$, wordt elke rij $\mathbf{h}_m$ van $\mathbf{H}$ door een achterwaartse vermenigvuldiging met $\mathbf{V}$ op zichzelf afgebeeld mits een zekere schaalfactor λ_m. Dit betekent tegelijkertijd dat elke kolom van $\mathbf{H}$ enkel een vermenigvuldiging met de diagonaalmatrix $\mathbf{\Lambda}$ ondergaat. Dit is de diagonaalmatrix die de overeenkomstige eigenwaarden λ_m bevat.

Definieer de matrix $\mathbf{H}'$ als

$$\mathbf{H}' = \mathbf{HV} \tag{5.6}$$

Kolom $\mathbf{h}'_n$ van $\mathbf{H}'$ is dan een lineaire combinatie van de k kolommen uit $\mathbf{H}$ die overeenkomen met de k dichtste buren van het punt dat overeenkomt met kolom $\mathbf{h}_n$ van $\mathbf{H}$. Indien de rijen van $\mathbf{H}$ eigenvectoren van $\mathbf{V}$ zijn, geldt er dat $\mathbf{h}'_n = \mathbf{\Lambda h}_n$. Dit betekent dat kolom $\mathbf{h}_n$ een sterke gelijkenis vertoont met de kolommen van zijn k dichtste buren. Punten met gemeenschappelijke buren zullen dus min of meer gelijke kolommen hebben in $\mathbf{H}$. Er kan dan geclusterd worden op basis van deze gelijkenissen.

Dit blijkt goed te werken op hetzelfde clusteringsprobleem als in figuur C-1. Indien de clusters minder duidelijk gescheiden zijn ondervindt deze methode echter problemen. Dit wordt geïllustreerd in figuur C-2 (zie kleurenbijlage). Hier krijgt elk punt een kleur op basis van de lineaire combinatie van vier basiskleuren, afhankelijk van de getallen in de corresponderende kolom van $\mathbf{H}$. De grote C-vormige cluster wordt in twee delen gesplitst. Herhalingen van dit experiment geven telkens gelijkaardige resultaten.

Er zijn bovendien enkele negatieve aspecten bij het toepassen van de eigenwaardenontbinding op dit

probleem:

- De kolommen van $\mathbf{H}$ zijn niet spaars. Dit betekent dat er in de meeste kolommen geen dominant element te vinden is die het mogelijk maakt om een classificatie-beslissing te nemen op basis van de kolom-index van dit element. De frames moeten dan geclassificeerd worden op basis van gelijkenis tussen de kolommen van $\mathbf{H}$. Uiteindelijk wordt er dus weinig bereikt, aangezien dit in wezen een herformulering van het oorspronkelijke probleem is. De kolommen van $\mathbf{H}$ kunnen namelijk opnieuw gezien worden als een set kenmerkenvectoren die geclassificeerd moeten worden.
- De matrix $\mathbf{H}$ bevat negatieve elementen. Negatieve gewichten zijn moeilijk interpreteerbaar, zoals reeds eerder werd aangehaald.
- Eigenwaarden en eigenvectoren kunnen complex zijn. Het is onduidelijk hoe complexe getallen moeten geïnterpreteerd worden bij de uiteindelijke classificatie.
- Een classificatie van de kolommen van $\mathbf{H}$ via een MLP blijkt op de TIMIT trainset slechts 54% correcte classificaties op te leveren[8]. Dit is een slechter resultaat dan de classificatieresultaten van NMF (zie tabel 5.1). Bovendien blijkt een eigenwaardenontbinding tot een overmodellering van de trainingsdata te leiden. Het MLP kan op een testset slechts 10,4% van de frames juist classificeren.

In de volgende sectie wordt een methode voorgesteld die op een eigenwaardenontbinding gebaseerd is, maar de bovenstaande problemen vermijdt.

5.4.2 'Positieve eigenwaardenontbinding' van de KNN-matrix

Het is wiskundig bewezen dat een willekeurige rijvector $\mathbf{h}^0$ via de iteratieprocedure

$$\mathbf{h}^{(t+1)} = \frac{\mathbf{h}^t\mathbf{V}}{\sum_i(\mathbf{h}^t\mathbf{V})_i} \tag{5.7}$$

naar een eigenvector behorende bij de dominante eigenwaarde van $\mathbf{V}^T$ convergeert. Indien $\mathbf{h}^0$ enkel positieve elementen bevat, zal steeds een eigenvector met positieve elementen gevonden worden. Deze iteratieprocedure is in eerste instantie onbruikbaar voor het opstellen van de matrix $\mathbf{H}$, aangezien elke rij van $\mathbf{H}$ naar dezelfde eigenvector zal convergeren. Er is een extra drijvende kracht nodig, die ervoor zorgt dat de rijen zo orthogonaal mogelijk zijn. De orthogonaliteitsvoorwaarde op de rijen van $\mathbf{H}$ kunnen vertaald worden naar voorwaarden op de 'spaarsheid' (*sparseness*) van de kolommen van $\mathbf{H}$.

De spaarsheid van een vector $\mathbf{x}$ kan gemeten worden op basis van de relatie tussen de L_1 norm en de L_2 norm van $\mathbf{x}$ [20]:

$$Sp(\mathbf{x}) = \frac{\sqrt{n} - \frac{\sum_i|x_i|}{\sqrt{\sum_i x_i^2}}}{\sqrt{n}-1} \tag{5.8}$$

waarbij n de dimensie is van $\mathbf{x}$. Deze functie evalueert tot 1 als en slechts als $\mathbf{x}$ slechts één component verschillend van nul bevat. Indien alle componenten gelijk zijn wordt de waarde van deze functie 0.

[8] Merk op dat er in dit geval geen matrix $\mathbf{W}$ beschikbaar is. Daarom wordt $\mathbf{W}$ in (5.4) voor alle MLP-experimenten van deze sectie gelijk gesteld aan $\mathbf{H}^T$. Het is namelijk arbitrair of $\mathbf{W}$ al dan niet $\mathbf{H}^T$ gebruikt worden als $\mathbf{W}$ in 5.4. Beide keuzes zijn verdedigbaar.

In [20] wordt een algoritme voorgesteld die de niet-negatieve vector zoekt die het dichtst gelegen is bij een gegeven vector (in Euclidische zin) en die een gegeven L_1 norm en L_2 norm heeft. Aan de hand van (5.8) kan dan een waarde voor de spaarsheid van een bepaalde vector opgelegd worden. In het vervolg van de tekst wordt dit algoritme het Hoyer-algoritme genoemd, naar de ontwerper ervan.

Dankzij het Hoyer-algoritme is het mogelijk om voorwaarden op de spaarsheid van de kolommen van $\mathbf{H}$ op te leggen. De matrix $\mathbf{H}$ wordt nu via de volgende methode opgesteld:

1. Initialiseer $\mathbf{H}$ als een positieve willekeurige matrix
2. $\mathbf{H}_{ij} \leftarrow \frac{\mathbf{H}_{ij}}{\sum_j \mathbf{H}_{ij}}$ voor elk element $\mathbf{H}_{ij}$
3. $\mathbf{H} \leftarrow \mathbf{HV}$
4. Pas het Hoyer-algoritme toe met opgegeven spaarsheid β op alle kolommen $\mathbf{h}_i$ van $\mathbf{H}$ waarvoor geldt dat $Sp(\mathbf{h}_i) < \beta$.
5. Ga terug naar 2 tenzij het aantal vooropgestelde iteraties werd bereikt.

In het vervolg van de tekst wordt deze methode een 'positieve eigenwaardenontbinding' van $\mathbf{V}^T$ genoemd, omwille van de gelijkenis met een eigenwaardenontbinding van de matrix $\mathbf{V}^T$. Er is echter geen éénduidige definitie van deze positieve eigenwaardenontbinding wegens de afhankelijkheid van de initialisatie en de parameter β.

Hoe hoger β gekozen wordt, hoe harder de uiteindelijke classificatie die door deze methode wordt gevonden. De normalisatie van de rijen van $\mathbf{H}$ in stap 2 vermijdt overflow. Dankzij deze normalisatie kunnen bovendien geen nulrijen ontstaan. De normalisatie zorgt er ook voor dat rijen met weinig dominante elementen een competitief voordeel krijgen t.o.v. rijen met veel dominante elementen. Op deze manier wordt vermeden dat één bepaalde rij alle dominante elementen in de matrix bevat. Dit laatste is belangrijk. Indien elke rij bv. genormaliseerd wordt via een deling met het maximale element in de rij, blijkt dat er 1 à 3 rijen ontstaan die alle dominante elementen bevatten.

Merk op dat de uiteindelijke matrix $\mathbf{H}$ steeds positief is. De kolommen van $\mathbf{H}$ zijn bovendien spaars (afhankelijk van de waarde van β), wat een belangrijke vereiste is om een eenvoudige classificaties te kunnen uitvoeren. Er is steeds convergentie van de getallen in de matrix $\mathbf{H}$. Er is op een artificiële dataset empirisch vastgesteld dat een matrix $\mathbf{H}$, die met een perfecte classificatie overeenkomt, niet verandert bij het toepassen van de bovenstaande procedure[9].

Hoewel deze methode over het algemeen geen perfecte classificatie vindt voor de artificiële dataset, zijn de resultaten beter dan die van het NMF-algoritme. In figuur C-3 (zie kleurenbijlage) worden 9 opeenvolgende resultaten van dergelijke experimenten getoond. Er werden steeds 200 iteraties uitgevoerd met $\beta = 0,1$. Daarna[10] worden 100 iteraties uitgevoerd met $\beta = 0,95$. Elk experiment verschilt in initialisatie. De spreiding van de punten is steeds gelijk. Vergelijk deze resultaten met de classificatie op basis van NMF (zie figuur C-1). De gevonden klassen zijn coherenter en minder verspreid over de verschillende puntenclusters in de ruimte. De clusters worden minder sterk opgedeeld in verschillende kleuren. In het algemeen wordt in ongeveer 15% van de gevallen een correcte classificatie bekomen (het experiment werd 40 keer uitgevoerd).

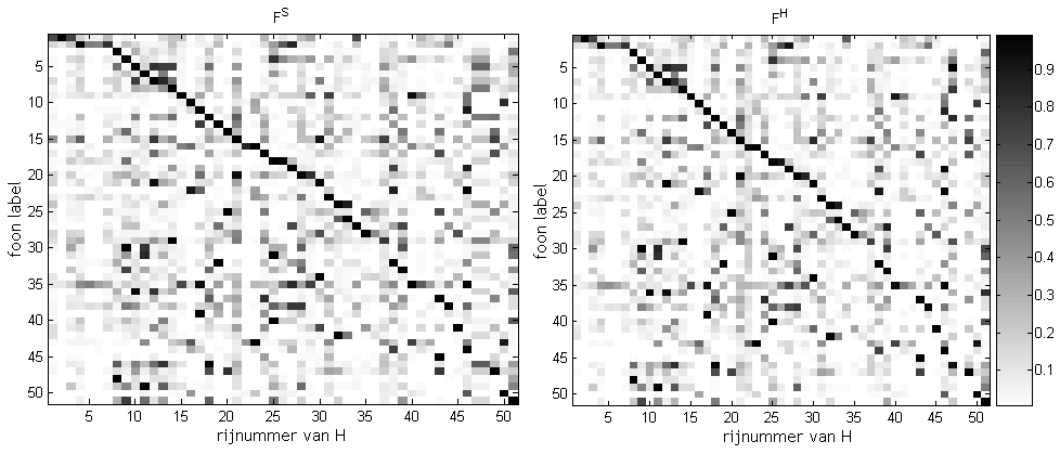
[9] De gebruikte dataset is dezelfde als deze van figuur C-2 in de kleurenbijlage.

[10] Uit experimenten blijkt dat betere resultaten bekomen worden indien eerst een groot aantal iteraties worden uitgevoerd met een lage waarde voor β. Daarna wordt β voor een aantal iteraties gelijkgesteld aan de vereiste spaarsheid. Kleine classificatiefouten kunnen nog weggewerkt worden door opnieuw te itereren met een lage waarde voor β, gevolgd door een aantal iteraties waarbij β gelijk is aan de vereiste spaarsheid.

5.4.3 Resultaten op de KNN-matrix met spraakdata

Het algoritme dat in de vorige sectie werd geïntroduceerd werd toegepast op de KNN-matrix met spraakdata. Er worden 200 iteraties uitgevoerd met $\beta = 0,1$. Daarna worden 80 iteraties uitgevoerd waarbij $\beta = 0,96$ wordt gekozen[11]. Hierna worden nog twee keer 80 iteraties uitgevoerd met een waarde voor β van respectievelijk $0,1$ en $0,96$. De rekentijd bedraagt ongeveer 100 seconden per iteratie als $\beta = 0,1$ en 120 seconden als $\beta = 0,96$ op een 2,4 GHz dual AMD opteron 280 processor met 16GB DDR400 geheugen.

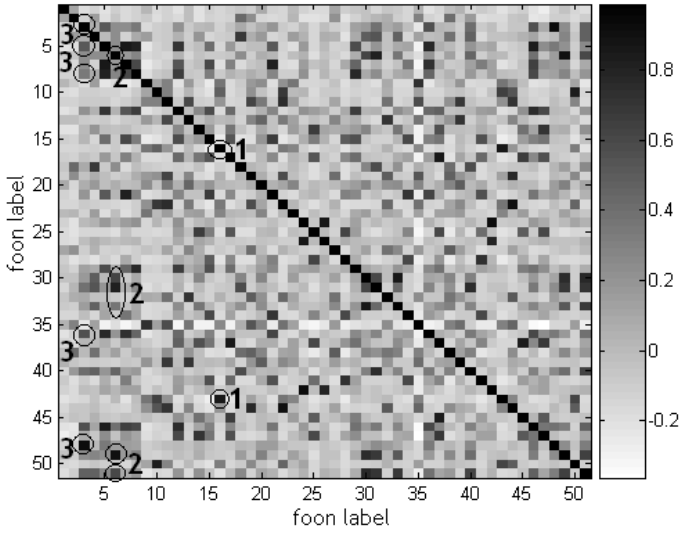
In figuur 5.5 worden de matrices $\mathbf{F}^{\mathrm{s}}$ en $\mathbf{F}^{\mathrm{h}}$ weergegeven, die op basis van de positieve eigenwaardenontbinding werden opgesteld. Ook nu blijkt het niet mogelijk te zijn om deze matrices te diagonaliseren. Bovendien blijkt de perfecte classificatie ook in dit geval geen stabiele oplossing te zijn. De classificatieresultaten met een MLP zijn 58,5% op de trainingset en 58,1% op de testset. Dit is gelijkaardig aan de classificatieresultaten op basis van NMF (zie tabel 5.1).



Figuur 5.5: De gediagonaliseerde matrices $\mathbf{F}^{\mathrm{s}}$ (links) en $\mathbf{F}^{\mathrm{h}}$ (rechts) na het uitvoeren van een positieve eigenwaardenontbinding van $\mathbf{V}^T$.

Een analyse van de correlatiecoëfficiënten tussen de rijen van deze matrix (zie figuur 5.6) toont opnieuw aan dat fonen, die gelijkaardig zijn in klank en uitspraak, door het algoritme ook vaak in dezelfde klasse worden ondergebracht. Merk op dat de rijen waartussen hoge correlaties bestaan, vaak ook in figuur 5.4 een hoge correlatiecoëfficiënt hebben. Ter illustratie werden enkele overeenkomstige elementen aangeduid in beide figuren. Ondanks het feit dat twee verschillende technieken werden gebruikt, bevatten de matrices met correlatiecoëfficiënten grote gelijkenissen. Dit is nogmaals een bevestiging van het vermoeden dat er een grote overlap is in de kenmerken-ruimte tussen bepaalde foonklassen met gelijkenissen in klank en uitspraak.

[11] Deze waarde voor β laat ongeveer twee dominante elementen toe per kolom van $\mathbf{H}$ indien de verticale dimensie $r = 51$. Eén daarvan zal echter steeds significant groter zijn dan de andere, wat interessant is in teken van een harde classificatie. Indien naar een zachte classificatie wordt gestreefd, kan $\beta = 0.93$ of lager gekozen worden.



Figuur 5.6: Correlatiecoëfficiënten tussen de rijen van de matrix $\mathbf{F}^h$ na het toepassen van een positieve eigenwaardenontbinding op de KNN-matrix

5.5 Besluit

Er werden twee methodes voorgesteld om een foon- of foneemclassificatie te vinden op basis van de KNN-matrix $\mathbf{V}$: NMF en de positieve eigenwaardenontbinding. Geen van beiden vindt een perfecte classificatie.

Er werd gewezen op enkele gebreken van de toegepaste methodes. Bij artificiële data slaagt NMF er niet in om een goede classificatie te vinden. De positieve eigenwaardenontbinding geeft over het algemeen betere classificaties en vindt, afhankelijk van de initialisatie, af en toe zelfs een perfecte classificatie.

De gebreken in beide methodes zijn vermoedelijk niet de hoofdoorzaak van het feit dat geen perfecte foonclassificatie wordt bereikt. Op artificiële datasets is een perfecte classificatie een stabiel punt van de updateformules van beide methodes. Op de spraakdata is dit echter niet het geval. Dit wijst erop dat bepaalde foonklassen vermoedelijk door elkaar liggen in de kenmerken-ruimte. Door deze overlap kan het gebruik van een KNN-matrix niet leiden tot een perfecte foonclassificatie. Ook de matrices met correlatiecoëfficiënten tussen de rijen van $\mathbf{F}^h$ bevestigen dit. Foonklassen die grote gelijkenissen hebben qua uitspraak en klank worden vaak door dezelfde latente klasse verklaard, aangezien de kenmerkenvectoren uit deze foonklassen te dicht bij elkaar liggen, of zelfs door elkaar liggen.

Hoofdstuk 6

Algemeen besluit

6.1 Samenvatting

Het vooropgestelde doel van dit eindwerk was om na te gaan of een computer in staat is om structuur te ontdekken in spraaksignalen, zonder gebruik te maken van extra informatie. Er werd gepoogd om dit via matrix-factorisatietechnieken op te lossen.

Een eerste doelstelling was om op basis van matrix-factorisatie een kenmerkenset te vinden ter vervanging van de MEL-filterbank. Hiervoor werden drie matrix-factorisatietechnieken getest: SWO, NMF met MSE-criterium en NMF met divergentie-criterium. Er werd een beter inzicht verworven in de eigenschappen van deze technieken. Ook de gebreken, zoals de energie-afhankelijkheid van het onderliggende kost-criterium en de gevolgen hiervan, werden uitgebreid toegelicht. Via NMF met divergentie-criterium werd een interessante factorisatie gevonden. De gevonden structuur blijkt grote gelijkenissen te vertonen met de MEL-filterbank en geeft ook gelijkaardige foneemherkenningsresultaten. Bovendien blijkt hieruit dat de productie van menselijke spraak zodanig geëvolueerd is dat de informatie, die nodig is om spraak te herkennen, zo goed mogelijk behouden blijft na analyse door het gehoorsysteem. Productie en analyse van spraak zijn blijkbaar goed aan elkaar aangepast.

Het tweede deelprobleem had als doelstelling om een automatische foneemclassificatie te vinden zonder gebruik te maken van a-priorische foneemkennis. Dit gebeurde via een factorisatie van een KNN-matrix, die opgesteld werd op basis van de kenmerkenvectoren van spraaksegmenten. Dit leverde een classificatie op die een zekere foneemkennis bevat. De kwaliteit van deze classificatie is echter onvoldoende. Er werd aangetoond waarom NMF geen ideale methode is voor een classificatie op basis van een KNN-matrix. Er werd bovendien een tweede methode geïntroduceerd die op artificiële datasets betere classificaties oplevert, namelijk de 'positieve eigenwaardenontbinding'. Een positieve eigenwaardenontbinding van de KNN-matrix met spraakdata vond echter geen foneemclassificatie met een hogere kwaliteit dan deze die via NMF werd gevonden. De oorzaak ligt vermoedelijk aan een te grote overlap van foneemklassen in de kenmerken-ruimte waardoor het KNN-model moeilijk tot een goede classificatie kan leiden.

Uit de experimenten van beide deelproblemen is gebleken dat het vinden van een makkelijk interpreteerbare structuur in spraaksignalen een moeilijk probleem is indien geen bijkomstige informatie mag gebruikt worden. De slaagkans hangt sterk af van de gebruikte factorisatiemethode, en van de voorverwerking van de data in de te factoriseren matrix.

6.2 Toekomstig onderzoek

De resultaten van de experimenten uit hoofdstuk 5 maken duidelijk dat het probleem van automatische foneemclassificatie nog niet volledig is opgelost. Er is nood aan een nieuw model ter vervanging van het KNN-model. Het zal waarschijnlijk nodig zijn om hiervoor ook tijdsinformatie te gebruiken. Zo kan bijvoorbeeld een HMM opgesteld worden waarbij het aantal toestanden gelijk is aan het aantal foon- of foneemklassen. Op deze manier worden ook overgangen tussen deze klassen expliciet gemodelleerd. Dit legt een striktere tijdsopdeling op. Indien de kansdichtheidsfuncties in de toestanden de dichtste buren van de geobserveerde vectoren voorspellen d.m.v. een Dirichlet verdeling, lijkt het schatten van de kansdichtheidsfuncties sterk op de in dit eindwerk onderzochte NMF-KNN combinatie. De schatting van de transitiekansen tussen de toestanden moet de observaties zo goed mogelijk verklaren. Dit kan gebeuren via een klassieke Baum-Welsch training.

De twee deelproblemen binnen dit eindwerk zijn slechts een klein aspect van het globale onderliggende doel om na te gaan of computers net zoals de mens in staat zijn om zelf een structuur te ontdekken in spraaksignalen. Gelijkaardige factorisatietechnieken kunnen ook toegepast worden op een hoger niveau zoals woorden, zinnen en zelfs over de zinsgrenzen heen. Om tot een globaal zelflerend model te komen op alle niveaus is er dus nog heel wat onderzoek nodig.

Bibliografie

[1] S. Young, "A review of large-vocabulary continuous-speech recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45–57, Sep. 1996.

[2] J. Holmes, *Speech Synthesis and Recognition*. Van Nostrand Reinhold (UK) Co. Ltd, 1988.

[3] K. Demuynck, J. Duchateau, D. Van Compernolle, and P. Wambacq, "Improved feature decorrelation for HMM-based speech recognition," in *Proc. International Conference on Spoken Language Processing*, vol. VII, Sydney, Australia, Dec. 1998, pp. 2907–2910.

[4] J. Duchateau, K. Demuynck, D. Van Compernolle, and P. Wambacq, "Class definition in discriminant feature analysis," in *Proc. European Conference on Speech Communication and Technology*, vol. III, Aalborg, Denmark, Sep. 2001, pp. 1621–1624.

[5] K. Demuynck, J. Duchateau, and D. Van Compernolle, "Optimal feature sub-space selection based on discriminant analysis," in *Proc. European Conference on Speech Communication and Technology*, vol. III, Budapest, Hungary, Sep. 1999, pp. 1311–1314.

[6] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[7] T. Landauer, P. Foltz, and D. Laham, "Introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.

[8] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence*, Stockholm, 1999.

[9] T. Hofmann and J. Puzicha, "Unsupervised learning from dyadic data," International Computer Science Insitute, Berkeley, CA, Tech. Rep. TR-98-042, 1998.

[10] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[11] E. Gaussier and C. Goutte, "Relation between plsa and nmf and implications," in *Proceedings of the ACM SIGIR conference on research and development in information retrieval*, Salvador, Brazil, 2005, pp. 601–602.

[12] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[13] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. DARPA Speech Recognition Workshop*, 1986, pp. 100–109.

[14] W. Fisher, V. Zue, J. Bernstein, and D. Pallett, "An acoustic-phonetic data base," *The Journal of the Acoustical Society of America*, vol. 81, no. S1, pp. S92–S93, May 1987.

[15] Z. Hafed and M. Levine, "Face recognition using the discrete cosine transform," *International Journal of Computer Vision*, vol. 43, no. 3, pp. 167–188, 2001.

[16] V. Eguiluz, M. Ospeck, Y. Choe, A. Hudspeth, and M. O. Magnasco, "Essential nonlinearities in hearing," *Physical Review Letters*, vol. 84, no. 22, pp. 5232–5235, 2000.

[17] D. S. Pallett, "Benchmark tests for darpa resource management database performance," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, Glasgow, UK, May 1989, pp. 536–539.

[18] R. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images," *Auditory Physiology and Perception, Proc. 9th International Symposium on Hearing*, 1992.

[19] K. Demuynck, "Extracting, modelling and combining information in speech recognition," Ph.D. dissertation, K.U.Leuven, ESAT, Feb. 2001.

[20] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *'Journal of Machine Learning Research'*, vol. 5, pp. 1457–1469, 2004.

Bijlage A

NMF met energie-onafhankelijke kostfunctie

In sectie 4.5.1 werd aangetoond dat het divergentie-criterium lineair afhankelijk is van de energie in de datamatrix $\mathbf{V}$. In deze sectie wordt een aanpassing van het NMF algoritme voorgesteld, opdat het kostcriterium onafhankelijk wordt van energie. Deze aanpassing is gebaseerd op de afleiding van de updateformules voor NMF volgens het divergentie-criterium zoals beschreven in [12].

Definitie A-1: $G(h, h')$ is een hulpfunctie voor $F(h)$ als de voorwaarden

$$G(h, h') \geq F(h), \quad G(h, h) = F(h) \tag{A-1}$$

voldaan zijn.

Lemma A-1: Als $G(h, h')$ een hulpfunctie is van $F(h)$, dan is $F(h)$ niet-stijgend door het toepassen van de update

$$h^{t+1} = \arg\min_{h} G(h, h^t) \tag{A-2}$$

Bewijs. $F(h^{t+1}) \leq G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$ □

$F(h^{t+1}) = F(h^t)$ geldt enkel als h^t een lokaal minimum is van $G(h, h^t)$. Aangezien F en G elkaar raken in dit punt zijn hun afgeleiden gelijk en is dit punt ook een stationair punt van F. Met andere woorden: een punt dat niet verandert onder de update (A-2), is een stationair punt van F.

De update-formules van het NMF-algoritme zijn op dit lemma gebaseerd. Indien de matrix $\mathbf{V}$ slechts uit 1 kolom v bestaat, geeft het divergentie-criterium de volgende kostfunctie:

$$F(h) = \sum_i \left[v_i \log \left(\frac{v_i}{\sum_a \mathbf{W}_{ia} h_a} \right) - v_i + \sum_a \mathbf{W}_{ia} h_a \right] \tag{A-3}$$

met $\mathbf{W}$ een vaste matrix met basisvectoren waarmee $\mathbf{V}$ wordt gereconstrueerd. De vector h bevat de gewichten voor deze basisvectoren om de vector v te reconstrueren. (A-3) bestaat uit een som van

termen die in de vorm $(\delta_i - \log(1+\delta_i))\, v_i$ kunnen geschreven worden (zie sectie 4.5.1). Hierbij is

$$\delta_i = \frac{\sum_a \mathbf{W}_{ia} h_a - v_i}{v_i} \tag{A-4}$$

de relatieve fout op de reconstructie van v_i. Indien v_i in elk van deze termen weggedeeld wordt, blijft enkel $(\delta_i - \log(1+\delta_i))$ over[1]. Dit is een functie van de relatieve fout. Op basis hiervan wordt een nieuwe kostfunctie $F'(h)$ gedefinieerd:

$$F'(h) = \sum_i \left[\log\left(\frac{v_i}{\sum_a \mathbf{W}_{ia} h_a} \right) - 1 + \frac{1}{v_i} \sum_a \mathbf{W}_{ia} h_a \right] \tag{A-5}$$

Deze nieuwe kostfunctie $F'(h)$ heeft als voordeel dat een gelijke kost wordt aangerekend voor gelijke relatieve reconstructiefouten op zowel hoge pieken als op lage pieken.

In [12] wordt bewezen dat

$$-\log \sum_a \mathbf{W}_{ia} h_a \leq -\sum_a \frac{\mathbf{W}_{ia} h_a^t}{\sum_b \mathbf{W}_{ib} h_b^t} \left(\log \mathbf{W}_{ia} h_a - \log \frac{\mathbf{W}_{ia} h_a^t}{\sum_b \mathbf{W}_{ib} h_b^t} \right) \tag{A-6}$$

Hieruit volgt dat

$$G'(h, h^t) = \sum_i \left[\log v_i - 1 + \sum_a \frac{1}{v_i} \mathbf{W}_{ia} h_a - \sum_a \frac{\mathbf{W}_{ia} h_a^t}{\sum_b \mathbf{W}_{ib} h_b^t} \left(\log \mathbf{W}_{ia} h_a - \log \frac{\mathbf{W}_{ia} h_a^t}{\sum_b \mathbf{W}_{ib} h_b^t} \right) \right] \tag{A-7}$$

een hulpfunctie is voor $F'(h)$.

Op basis van deze hulpfunctie kan de volgende stelling bewezen worden:

Stelling A-1:

1. De kostfunctie $F'(h)$ is niet-stijgend onder de update

$$h_a^{t+1} = \frac{h_a^t}{\sum_i \frac{\mathbf{W}_{ia}}{v_i}} \sum_i \frac{\mathbf{W}_{ia}}{\sum_b \mathbf{W}_{ib} h_b^t} \tag{A-8}$$

2. Een punt h_0 dat invariant is onder de bovenstaande update, is een stationair punt van $F'(h)$.

Bewijs. Het minimum van $G'(h, h^t)$ met betrekking tot h wordt bepaald door de gradiënt gelijk te stellen aan nul:

$$\frac{\partial G'}{\partial h_a} = -\sum_i \frac{\mathbf{W}_{ia} h_a^t}{\sum_b \mathbf{W}_{ib} h_b^t} \frac{1}{h_a} + \sum_i \frac{\mathbf{W}_{ia}}{v_i} = 0 \tag{A-9}$$

Dit uitwerken naar h_a geeft

$$h_a = \frac{h_a^t}{\sum_i \frac{\mathbf{W}_{ia}}{v_i}} \sum_i \frac{\mathbf{W}_{ia}}{\sum_b \mathbf{W}_{ib} h_b^t} \tag{A-10}$$

Het punt h_a gedefinieerd volgens (A-10) is een minimum van $G'(h, h^t)$. Via lemma A-1 en de gevolgen van dit lemma zijn beide onderdelen van de stelling bewezen. □

[1] Deze functie kan beschouwd worden als een heel ruwe benadering van een lineaire functie van de relatieve fout δ_i indien δ_i positief is. De benadering gaat ervan uit dat de log-term kan verwaarloosd worden t.o.v. de lineaire term als $\delta_i > 0$. Figuur 4.10 geeft dit weer. Het is duidelijk dat de voorkeur uitgaat naar overschattingen (δ_i positief). Volgens dezelfde redenering kan het divergentie-criterium beschouwd worden als een benadering van een lineaire functie van de absolute fout op de reconstructie.

In het geval dat $\mathbf{V}$ uit meerdere kolommen bestaat, kan de updateformule (A-8) geschreven worden als een matrix-updateformule:

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{\sum\limits_{i} \mathbf{W}_{ia} / (\mathbf{WH})_{i\mu}}{\sum\limits_{k} \frac{\mathbf{W}_{ka}}{\mathbf{V}_{k\mu}}} \tag{A-11}$$

Door de rollen van $\mathbf{W}$ en $\mathbf{H}$ om te draaien kan de updateregel voor $\mathbf{W}$ afgeleid worden:

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{\sum\limits_{\mu} \mathbf{H}_{a\mu} / (\mathbf{WH})_{i\mu}}{\sum\limits_{n} \frac{\mathbf{H}_{an}}{\mathbf{V}_{in}}} \tag{A-12}$$

Bijlage B

Lijst met labels van fonen

Hieronder staat de genummerde lijst van fonen die elk een label krijgen in de manuele transcriptie van de TIMIT databank. De nummers komen overeen met de rij-indices van de matrices $\mathbf{F}^s$ en $\mathbf{F}^h$. De symbolen zijn rechtstreeks overgenomen uit de TIMIT-transcriptie. Dit zijn notaties uit het 'JAPA' (just another phonetic alphabet) met enkele Engelse uitbreidingen.

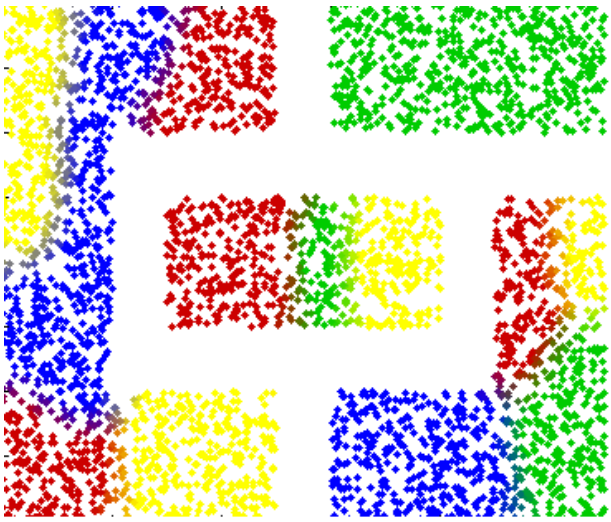
1.	[#]	21.	[E:]	41.	[A+w]
2.	[%0]	22.	[t+S]	42.	[,r!]
3.	[i]	23.	[%]	43.	[z]
4.	[v]	24.	[,r]	44.	[d+Z]
5.	[I!]	25.	[O]	45.	[,*l*]
6.	[n]	26.	[r]	46.	[u]
7.	[E]	27.	[b]	47.	[O+j]
8.	[I]	28.	[A]	48.	[j]
9.	[f]	29.	[@]	49.	[,n]
10.	[S]	30.	[,m]	50.	[Z]
11.	[t]	31.	[N]	51.	[,N]
12.	[u!]	32.	[g]		
13.	[k]	33.	[m]		
14.	[w]	34.	[A+j]		
15.	[@!]	35.	[T]		
16.	[s]	36.	[E+j]		
17.	[O+w]	37.	[*l*]		
18.	[%1]	38.	[D]		
19.	[d]	39.	[p]		
20.	[h]	40.	[d1]		

Hierin staat [#] voor een stilteframe, [%0] voor een closure van een stemloos plosief, [%1] voor een closure van een stemhebbend plosief, [d1] voor een geaspireerde d (zoals in 'butter') en [%] voor een 'epithetic closure'.

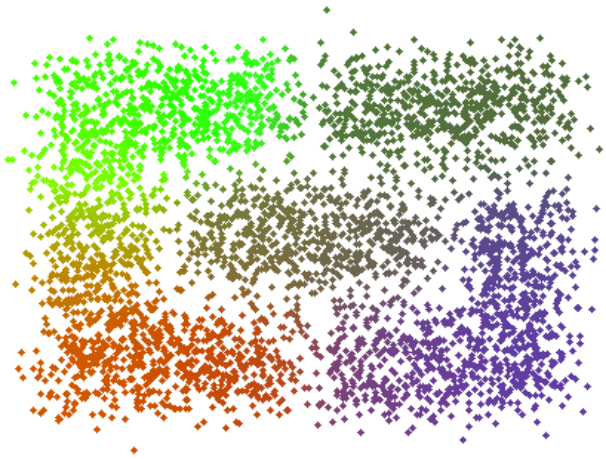
Een '!' betekent dat dit een klank met een klemtoon is. Een komma betekent dat het om een syllabische foon gaat (zoals op het einde van 'button', 'father').

Bijlage C

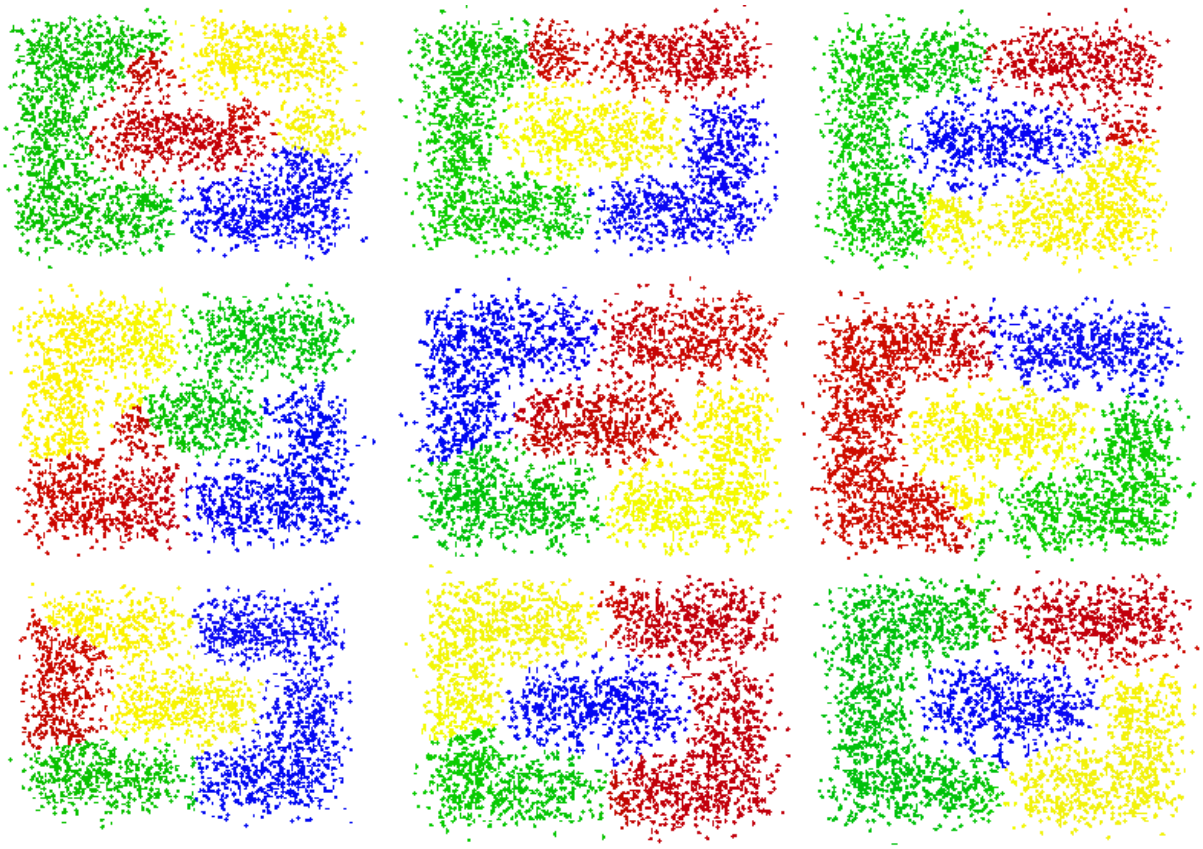
Kleurenbijlage



Figuur C-1: Classificatie door NMF op een artificieel voorbeeld met vier duidelijk gescheiden klassen. Punten met gelijke kleuren worden door NMF tot dezelfde klasse gerekend



Figuur C-2: Classificatie via eigenwaardenontbinding op een artificieel voorbeeld met vier niet perfect gescheiden klassen. Punten met gelijkaardige kleuren hebben gelijkaardige corresponderende kolommen in $\mathbf{H}$.



Figuur C-3: 9 opeenvolgende classificatie-experimenten op basis van een ‘positieve eigenwaardenontbinding’