

EstiHMM: een efficiënt algoritme
ter bepaling van de maximale sequenties
in een imprecies hidden Markovmodel

Jasper De Bock

Promotor: prof. dr. ir. Gert De Cooman

Begeleiders: ir. Filip Hermans, dr. ir. Erik Quaeghebeur

Masterproef ingediend tot het behalen van de academische graad van
Master in de ingenieurswetenschappen: bouwkunde

Vakgroep Elektrische energie, Systemen en Automatisering

Voorzitter: prof. dr. ir. Jan Melkebeek

Faculteit Ingenieurswetenschappen en Architectuur

Academiejaar 2010–2011



*Opgedragen aan iedereen die mij
heeft blij gemaakt*

Voorwoord

De vele uitdagingen die deze masterproef mij heeft geboden, de competente, steeds tot hulp bereid zijnde en – niet te vergeten – vriendschappelijke omgeving (Dankjewel aan de voltallige onderzoeksgroep SYSTeMS!) waarin ik deze uitdagingen bij de horens kon vatten en de voldoening en blijheid die dit eindresultaat mij schenkt, kunnen moeilijk naar waarde omschreven worden. Ik doe een poging:

“Wauw, dit was godverdomme plezant!”

Er zijn vele mensen zonder wie deze masterproef en de ermee gepaard gaande vreugde nooit hun huidige vorm hadden kunnen aannemen. Ik wil hen graag bedanken.

Gert, voor alles. . . en dat is verdorie veel! Voor de kansen die hij mij gegeven heeft. Voor zijn intense, zeer competente begeleiding. Voor de manier waarop hij zijn passie voor en enthousiasme over zijn vakgebied heeft weten over te brengen op mij. Voor de ongelofelijke hoeveelheid tijd die hij heeft vrijgemaakt voor de begeleiding van deze masterproef. En bovenal voor de informele, vriendschappelijke sfeer waarin dit alles kon gebeuren. Het was me dan ook een eer en een genoegen om hem als mijn promotor te hebben. Ik weet niet hoe ik hem hier ooit voldoende kan voor bedanken, maar een Westmalle Tripel lijkt me alvast een goed begin.

De begeleiders van deze masterproef, Erik en Filip, ook voor alles. Men kan zich geen betere begeleiding wensen. Ze spendeerden een enorme hoeveelheid tijd aan het nalezen van teksten, beantwoorden van vragen en oplossen van allerlei praktische problemen. Ik ben hen hier zeer dankbaar voor. Ik wil hen ook bedanken voor de interessante en leuke babbels en voor de vele grappige momenten. Het was een plezier om met hen samen te werken.

Marco, omdat hij mij de kans gaf het EstiHMM-algoritme, toen het nog in zijn kinderschoenen stond, voor te stellen in zijn onderzoeksgroep. Hij gaf mij zo de gelegenheid om kennis te maken met onderzoek in een internationale context. Ik wil hem ook, en vooral, bedanken voor de hartelijke manier waarop hij mij daar ontvangen heeft.

Arthur, mijn lotgenoot in het grote masterproef-avontuur. Zonder hem had ik misschien nooit dit masterproefonderwerp ontdekt. Hij was het die me als eerste over dit onderwerp vertelde en dacht dat het misschien iets voor mij was. Hij bleek gelijk te hebben. Ik wil hem ook bedanken voor een jaar van intense, zeer leuke samenwerking. Voor de koffietjes, de boeiende gesprekken en de pauzes tussen het vele schrijven in. En vooral voor zijn immer vrolijke aanwezigheid. Deze masterproef is opgedragen aan iedereen die mij heeft blij gemaakt en hij staat met recht en rede ergens bovenaan die lijst.

Behalve de neerslag van een jaar onderzoek, is dit ook het einde van mijn opleiding en van een levensfase. Dat ik na 22 jaar hier beland ben, heb ik aan vele mensen te danken. Ook aan hen wil ik hier een dankwoord richten.

Mijn beste vrienden, omdat in hun definitie van vriendschap het woord “Facebook” niet voorkomt. Omdat zij – in tegenstelling tot sommige andere mensen – weten dat vriendschap verdiend wordt en niet met een muisklik aan of uit kan gezet worden. Omdat ze er voor me zijn als ik ze nodig heb. Omdat ze mij kennen en aanvaarden, in al mijn facetten. Bedankt, jullie hebben een mooier mens van me gemaakt.

Annelien, voor haar liefde. Ze heeft mij veel geleerd over één van de meest complexe dingen die ik ken. Ik koester de momenten van gelukzaligheid, die ik aan haar te danken heb. Liefde is niet alles, maar zonder liefde is alles niets. . .

Mijn familie, omdat ik een mengelmoes ben van hen allemaal. Ze dienen mij als voorbeeld: soms omdat ik hun fouten niet wil herhalen, soms omdat ik ooit hoop te kunnen wat zij kunnen en meestal beide tegelijk. Ik zie hen graag en hoop nog lang het geluk te hebben ze dicht bij mij te kunnen houden.

Mijn vader, omdat hij me – zowel genetisch als via m’n opvoeding – mijn liefde voor wetenschap heeft meegegeven. Omdat hij me kritisch naar deze wereld heeft leren kijken en zo mijn overlevingskansen op deze grote blauwe bol sterk vergroot heeft.

Mijn moeder, omdat ze weet dat het belangrijkste ingrediënt voor een goede opvoeding, liefde is. En omdat ze mij die gegeven heeft, onvoorwaardelijk, zoals alleen moeders dat kunnen. Een kind kan zich geen betere moeder wensen dan de mijne. Ze zal – net als ik – met lichte weemoed afscheid nemen van mijn kindertijd, maar ze mag gerust zijn: ik kom voorlopig nog naar huis om mijne was te doen. . .

Toelating tot bruikleen

De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.

Jasper De Bock
6 juni 2011

Overzicht

MASTERPROEF ingediend tot het behalen van de academische graad van burgerlijk bouwkundig ingenieur

Titel: *EstiHMM: een efficiënt algoritme ter bepaling van de maximale sequenties in een imprecies hidden Markovmodel*

Auteur: *Jasper De Bock*

Promotor: *prof. dr. ir. Gert De Cooman*

Begeleiders: *ir. Filip Hermans, dr. ir. Erik Quaeghebeur*

Onderzoeksgroep SYSTEMS

Directeur: *prof. dr. ir. Dirk Aeyels*

Vakgroep Elektrische energie, systemen en Automatisering

Voorzitter: *prof. dr. ir. Jan Melkebeek*

Faculteit Ingenieurswetenschappen en Architectuur

Universiteit Gent

Academiejaar 2010-2011

Samenvatting

In deze masterproef ontwikkelen we het EstiHMM-algoritme (Estimation in imprecise Hidden Markov Models). Dit algoritme bepaalt op efficiënte wijze de maximale sequenties in een imprecies hidden Markovmodel. In het precieze geval kan dit probleem makkelijk opgelost worden met het gekende Viterbi-algoritme. In het imprecieze geval bestond er tot op heden echter geen efficiënte oplossing.

Om de meerwaarde aan te tonen van het EstiHMM-algoritme, ten opzichte van het Viterbi-algoritme, voeren we enkele experimenten uit en behandelen we een concrete toepassing. Uit onze resultaten blijkt dat het EstiHMM-algoritme er in slaagt eventuele robuustheidsproblemen van het Viterbi-algoritme aan te geven en op te vangen. In situaties waar een model moet opgebouwd worden aan de hand van een beperkte hoeveelheid data, kan een dergelijke robuuste aanpak een grote meerwaarde bieden.

De toepassingsmogelijkheden van het EstiHMM-algoritme zijn legio. Elke toepassing van het Viterbi-algoritme kan namelijk ook behandeld worden met het EstiHMM-algoritme. Mogelijke toepassingsdomeinen zijn bijvoorbeeld spraakherkenning, bio-informatica, optische tekenherkenning en cryptanalyse.

Trefwoorden

imprecies hidden Markovmodel – coherente onderprevisie – epistemische irrelevantie – maximale toestandssequentie – Viterbi-algoritme

EstiHMM: an efficient algorithm for state sequence prediction in imprecise hidden Markov models

Jasper De Bock

Supervisor: prof. dr. ir. Gert De Cooman

Abstract— We develop an efficient algorithm that calculates the maximal state sequences in an imprecise hidden Markov model by means of coherent lower previsions. Initial results show that this algorithm is able to robustify the inferences made by a classical precise model.

Keywords— Imprecise hidden Markov model, coherent lower prevision, epistemic irrelevance, maximal state sequence, Viterbi algorithm.

I. INTRODUCTION

WE present an efficient exact algorithm for estimating state sequences from outputs (or observations) in imprecise hidden Markov models (iHMM), where both the uncertainty linking one state to the next, and that linking a state to its output, are represented using coherent lower previsions. The notion of independence we associate with the credal network representing the iHMM is that of epistemic irrelevance. We consider as best estimates for state sequences the (Walley–Sen) maximal sequences [1] for the posterior joint state model (conditioned on the observed output sequence), associated with a gain function that is the indicator of the state sequence.

Our algorithm corresponds to (and generalises) finding the state sequence with the highest posterior probability in HMMs with precise transition and output probabilities, for which the existing Viterbi-algorithm [2] provides an efficient solution. However, for imprecise-probabilistic local models, such as coherent lower previsions, we know of no algorithm in the literature for which the computational complexity comes even close to that of Viterbi. We solve this problem by developing an algorithm which has a computational complexity that is at worst quadratic in the length of the Markov chain, cubic in the number of states, and essentially linear in the number of maximal state sequences.

II. COHERENT LOWER PREVISIONS

The theory of coherent lower previsions was the most important mathematical tool to develop this algorithm. We refer to [3] for an in-depth study of this theory.

Coherent lower previsions are a way of working with imprecise probabilities. Whereas classical probability theory assumes that a model can be represented by a single probability mass function, the theory of imprecise probabilities works with a sets of possible probability mass functions and therefore allows for imprecision to be modelled. For people who have never heard of this notion, looking at it as a way of robustifying the classical theory is perhaps the most useful interpretation.

Consider a set \mathcal{M} of probability mass functions on \mathcal{X} . With each $p \in \mathcal{M}$, we can associate a linear prevision P_p , defined on the set $\mathcal{L}(\mathcal{X})$ of all real-valued maps on \mathcal{X} . For every $f \in \mathcal{L}(\mathcal{X})$, $P_p(f)$ is the expected value of f , according to the probability mass function p . We then define the lower prevision $\underline{P}_{\mathcal{M}}$ that corresponds with \mathcal{M} as follows:

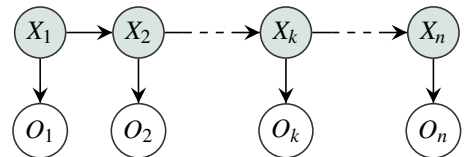
$$\underline{P}_{\mathcal{M}}(f) = \min \{P_p(f) : p \in \mathcal{M}\}.$$

If \mathcal{M} is closed and convex (which makes it a credal set), it can be shown [3] that $\underline{P}_{\mathcal{M}}$ satisfies a set of interesting mathematical properties and we call $\underline{P}_{\mathcal{M}}$ a coherent lower prevision. Furthermore, it is proven in [3] that such coherent lower previsions are completely equivalent with their corresponding credal set. This implies that we can use the theory of coherent lower previsions as a tool for reasoning with closed convex sets of probability mass functions. From now on, we will not indicate the credal set \mathcal{M} anymore and we will simply talk about coherent lower previsions \underline{P} , but one should keep in mind that there is always a credal set that corresponds with such a coherent lower prevision.

In a similar way, we can also define conditional lower previsions, which are an extension of the classical conditional expectation functional. More information about these conditional lower previsions can be found in [3].

III. IMPRECISE HIDDEN MARKOV MODELS

An imprecise hidden Markov model can be depicted using the following probabilistic graphical model:



The *state variables* X_1, \dots, X_n assume values in the respective finite sets $\mathcal{X}_1, \dots, \mathcal{X}_n$, and the *output variables* O_1, \dots, O_n assume values in the respective finite sets $\mathcal{O}_1, \dots, \mathcal{O}_n$. We denote generic values of X_k by x_k or \hat{x}_k and generic values of O_k by o_k . If we talk about complete sequences of variables, we will use the index $1 : n$ instead of k . For example, $O_{1:n}$ is the *output sequence*, which takes values $o_{1:n}$ in the set $\mathcal{O}_{1:n}$.

The local uncertainty models associated with the nodes of the network are (conditional) coherent lower previsions, and the independence notion used to interpret the graphical structure is that of epistemic irrelevance [3], [4].

We show how we can use the ideas in [5] (independent natural extension and marginal extension) to construct a most conservative joint model \underline{P}_1 from the imprecise local transition and emission models, and derive a number of interesting and useful formulas from that construction.

IV. THE PROBLEM TO BE SOLVED

In a hidden Markov model, the states are not directly observable, but the outputs are, and the general aim is to use the outputs to estimate the states. We concentrate on the following problem: *Suppose we have observed the output sequence $o_{1:n}$, estimate the state sequence $x_{1:n}$.*

The first step in our approach consists in updating (or conditioning) the joint model \underline{P}_1 on the observed outputs $O_{1:n} = o_{1:n}$. We do this by using the so-called regular extension [3]:

$$\underline{P}_1(f|o_{1:n}) = \max \{ \mu \in \mathbb{R} : \underline{P}_1(\mathbb{I}_{\{o_{1:n}\}}[f - \mu]) \geq 0 \},$$

for all $f \in \mathcal{L}(\mathcal{X}_{1:n})$.

The next step consists in using the posterior model $\underline{P}_1(\cdot|o_{1:n})$ to find best estimates for the state sequence $x_{1:n}$. On the Bayesian approach, this is usually done by solving a decision-making, or optimisation, problem: we associate a gain function $\mathbb{I}_{\{x_{1:n}\}}$ with every candidate state sequence $x_{1:n}$, and select as best estimates those state sequences $\hat{x}_{1:n}$ that maximise the expected gain, resulting in state sequences with maximal posterior probability.

Here we generalise this decision-making approach towards working with imprecise probability models. The criterion we use to decide which estimates are optimal for the given gain functions is that of (Walley–Sen) *maximality* [1], [3].

We prove (for the specific case of our interpretation of an iHMM) that the collection $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ of all optimal state sequences $\hat{x}_{1:n}$ can be defined as follows:

$$\begin{aligned} \hat{x}_{1:n} \in \text{opt}(\mathcal{X}_{1:n}|o_{1:n}) \\ \Leftrightarrow (\forall x_{1:n} \in \mathcal{X}_{1:n}) \underline{P}_1(\mathbb{I}_{\{o_{1:n}\}}[\mathbb{I}_{\{x_{1:n}\}} - \mathbb{I}_{\{\hat{x}_{1:n}\}}]) \leq 0. \end{aligned}$$

In summary then, we develop an efficient algorithm for finding the set of maximal estimates $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$.

V. THE ESTIHMM-ALGORITHM

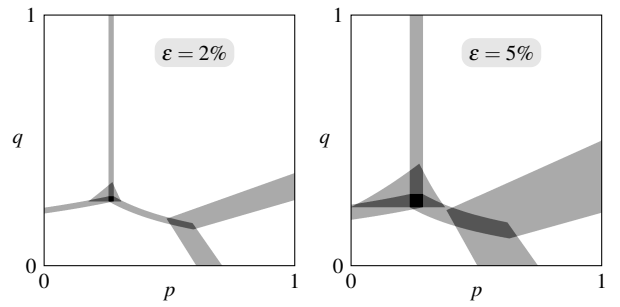
Finding all maximal state sequences in $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ seems a daunting task at first: it has a search space that grows exponentially in the length of the Markov chain. However, we have been able to use the basic formulas mentioned in the section on iHMMs to derive an appropriate version of Bellman’s [6] Principle of Optimality, which allows for an exponential reduction of the search space. By using a number of additional tricks, we are then able to devise an algorithm for finding all maximal state sequences that is essentially linear in the number of such maximal sequences, quadratic in the length of the chain, and cubic in the number of states. We have given this algorithm the name EstiHMM: Estimation in imprecise Hidden Markov Models.

We perceive the complexity of the EstiHMM algorithm to be comparable to that of the Viterbi algorithm, especially after realising that the latter makes the simplifying step of resolving ties more or less arbitrarily in order to produce only a single optimal state sequence. This is something we will not allow our algorithm to do, as this would completely remove the advantage of robustness our algorithm offers.

VI. SOME EXPERIMENTS

Using the EstiHMM algorithm, we have calculated the number of maximal sequences corresponding to a precise model that has been contaminated to make it imprecise. The degree of imprecision is expressed by the parameter ε . In the figures below, the number of maximal sequences is depicted as a function of the precise transition probabilities p and q for two different degrees of imprecision. Darker areas correspond to more maximal sequences. We observe that the areas with more than one maximal

sequence enlarge as the imprecision grows and show that they are expanded versions of the lines of indifference that occur in the precise case.



VII. AN APPLICATION

As a first and simple application, we use the EstiHMM algorithm to try and correct mistakes in words that have been processed by Optical Character Recognition software (OCR). We compare our results with those of the Viterbi algorithm and show that our algorithm offers a more robust solution.

For example, the word CHE was wrongfully read as CNE by the OCR software. Using a precise model, the Viterbi algorithm could not correct this mistake, as it suggested that the original correct word was ONE. The EstiHMM algorithm on the other hand, using an imprecise model, concluded that it was not sure enough to offer a single solution and suggested CBE, CHE, CNE, CZE and ONE as possible solutions, thereby including the correct one.

VIII. CONCLUSIONS

Interpreting the graphical structure of an imprecise hidden Markov model as a credal network under epistemic irrelevance, leads to an efficient algorithm for finding the maximal state sequences for a given output sequence. Preliminary experiments and a first simple application show that this algorithm extends the well-known Viterbi algorithm in such a way that it allows for imprecision and that it can be used to robustify the inferences made by a classical precise model.

ACKNOWLEDGMENTS

The author would like to acknowledge the huge amount of time, help and knowledge that Gert de Cooman, Erik Quaeghebeur and Filip Hermans have offered him during his research.

REFERENCES

- [1] Matthias C. M. Troffaes, “Decision making under uncertainty using imprecise probabilities,” *International Journal of Approximate Reasoning*, vol. 45, no. 1, pp. 17–29, January 2007.
- [2] Andrew J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [3] Peter Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, London, 1991.
- [4] Inés Couso, Serafin Moral, and Peter Walley, “A survey of concepts of independence for imprecise probabilities,” *Risk Decision and Policy*, vol. 5, pp. 165–181, 2000.
- [5] Gert de Cooman, Filip Hermans, Alessandro Antonucci, and Marco Zaffalon, “Epistemic irrelevance in credal nets: the case of imprecise Markov trees,” *International Journal of Approximate Reasoning*, vol. 51, pp. 1029–1052, 2010.
- [6] Richard Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.

Inhoudsopgave

1	Inleiding	1
1.1	Probleemschets	1
1.2	Bijdrage van deze masterproef	2
1.3	Opbouw van deze masterproef	2
1.4	Over de taal in dit werk	5
2	Markovketens	6
2.1	Inleiding	6
2.2	Toestandsveranderlijken	7
2.3	Een model op basis van waarschijnlijkheden	8
2.3.1	Lokale modellen	8
2.3.2	Een globaal model	9
2.4	De Markovvoorwaarde	10
2.5	Stationariteit	11
2.6	Een model op basis van verwachtingswaarden	12
2.6.1	Lokale modellen	12
2.6.2	Een globaal model	13
2.7	Grafische voorstelling	14
3	Hidden Markovmodellen	16
3.1	Inleiding	16
3.2	Observaties en verborgen toestanden	17
3.3	Opbouw van een globaal model	18
3.4	Het Viterbi-algoritme	20
3.4.1	Probleemstelling	20
3.4.2	Twee verschillende versies van het Viterbi-algoritme	22
3.4.3	Een versie die alle oplossingen bepaalt	23
3.4.4	Een versie die slechts één oplossing bepaalt	25
3.4.5	Bespreking van de complexiteit	26
3.4.6	Implementatie in Python	27

4	Coherente onderprevisies	28
4.1	Inleiding	28
4.2	Imprecieze waarschijnlijkheden	29
4.3	Onderprevisies	30
4.4	Coherentie	31
4.5	Terug naar credale verzamelingen	33
4.6	Conditionele onderprevisies	35
4.7	Reguliere extensie	36
4.7.1	Een bevattelijke definitie	36
4.7.2	Een handiger uitdrukking	36
4.7.3	Enkele eigenschappen	38
5	Imprecieze hidden Markovmodellen	40
5.1	Inleiding	40
5.2	Hetzelfde, maar “met streepjes eronder”	40
5.3	Lokale modellen	41
5.4	Interpretatie van de grafische structuur	42
5.5	Recursieve opbouw	43
5.6	Enkele handige eigenschappen	45
6	Het EstiHMM-algoritme	48
6.1	Inleiding	48
6.2	Conditioneren van het globaal model	49
6.3	Maximale sequenties	51
6.4	Nuttige recursiebetrekkingen	53
6.5	Optimaliteitsprincipe	54
6.6	Voorafgaande berekeningen	56
6.7	Herformulering van de optimaliteitsvoorwaarde	58
6.8	Het algoritme	60
6.8.1	Een recursieve oplossing	60
6.8.2	Een efficiënte methode voor de essentiële stap	60
6.8.3	De complexiteit van de essentiële stap	62
6.9	Bespreking van de complexiteit van het algoritme	66
6.10	Implementatie in Python	67
7	Enkele experimenten	68
7.1	Inleiding	68
7.2	Analyse van het aantal maximale elementen	68
7.2.1	Beschrijving van een binaire, stationaire iHMM	68
7.2.2	De basisprincipes voor een sequentie met lengte twee	69
7.2.3	Resultaten voor een sequentie met lengte drie	72
7.3	Een illustratie van de efficiëntie van het algoritme	73

8 Een toepassing	75
8.1 Inleiding	75
8.2 Omschrijving van het probleem	75
8.3 De gebruikte data	76
8.4 Observaties en verborgen toestanden	78
8.5 Modelling met een HMM	78
8.5.1 Lokale precieze modellen opstellen	78
8.5.2 Resultaten van het Viterbi-algoritme	79
8.6 Modelling met een iHMM	83
8.6.1 Lokale imprecieze modellen opstellen	83
8.6.2 Resultaten van het EstiHMM-algoritme	84
8.6.3 Verklaring van vreemde oplossingen	87
8.6.4 Robuustheid van het EstiHMM-algoritme	89
8.6.5 Analyse van het aantal oplossingen	91
9 Besluit	94
A Bewijzen	96
B Data Divina Commedia	101
B.1 Originele modeltekst	101
B.2 Genormaliseerde originele modeltekst	104
B.3 Modeltekst na inlezing door OCR-programma	106
B.4 Woordenlijst modeltekst	108
B.5 Analyse van de modeltekst	112
B.5.1 Beginletters van de originele woorden	112
B.5.2 Overgangen tussen letters in de originele woorden	112
B.5.3 Door het OCR-programma gemaakte fouten	113
B.6 Lokale modellen	114
B.6.1 Marginaal model	114
B.6.2 Transitie-model	115
B.6.3 Emissie-model	116
B.7 Originele voorbeeldtekst	117
B.8 Genormaliseerde originele voorbeeldtekst	118
B.9 Voorbeeldtekst na inlezing door OCR-programma	118
B.10 Woordenlijst voorbeeldtekst	119
B.11 Resultaten van EstiHMM en Viterbi	120
C Python-code	127
C.1 Viterbi-algoritme	127
C.2 EstiHMM-algoritme	132
C.3 Gemeenschappelijke code	139
Referentielijst	144

Hoofdstuk 1

Inleiding

“De enige plaats waar succes eerder komt dan werk, is het woordenboek.”

– Vidal Sassoon

1.1 Probleemschets

Hidden Markovmodellen worden reeds lange tijd met succes toegepast in heel wat domeinen. Enkele voorbeelden zijn spraakherkenning, bio-informatica, optische tekenherkenning en cryptoanalyse. Een belangrijk probleem hierbij is het vinden van een verborgen sequentie van veranderlijken, uitgaande van een waargenomen sequentie van observaties. Men kan bijvoorbeeld op zoek zijn naar de inhoud van een geheim bericht en enkel beschikken over de gecodeerde versie.

Een zeer efficiënte methode om dit probleem op te lossen is het gekende Viterbi-algoritme. Dit algoritme berekent uitgaande van een model en een reeks observaties de maximale verborgen toestandssequentie. Dit is de toestandssequentie die, gegeven de waargenomen observaties, de grootste waarschijnlijkheid heeft om opgetreden te zijn. Het Viterbi-algoritme is eenvoudig en vlot toepasbaar en haar resultaten worden in de bovenstaande domeinen veelvuldig gebruikt om heel wat praktische problemen op te lossen.

Een tekortkoming van dit algoritme is dat het veronderstelt dat we over een precies model beschikken dat de onderzochte situatie modelleert. Een dergelijke veronderstelling is in vele gevallen niet te verantwoorden. We beschikken bijvoorbeeld vaak slechts over een te beperkte hoeveelheid data om een betrouwbaar precies model te kunnen opbouwen. Het zou in die gevallen nuttig zijn om geen precies, maar een imprecies hidden Markovmodel te beschouwen. Een dergelijk model kan wel op coherente en betrouwbare wijze rekening houden met een eventueel gebrek aan voldoende data en zal daarom veel robuustere resultaten opleveren. Tot op heden bestond er voor imprecieze hidden Markovmodellen echter geen equivalent van het Viterbi-algoritme.

1.2 Bijdrage van deze masterproef

Veruit de belangrijkste bijdrage van deze masterproef bestaat uit de ontwikkeling van het EstiHMM¹-algoritme. Dit algoritme bepaalt op efficiënte wijze de maximale sequenties in een imprecies hidden Markovmodel en lost hiermee het in de vorige paragraaf beschreven probleem op.

Een tweede, minder belangrijke bijdrage is de beschrijving van een variant van het Viterbi-algoritme die algemener is dan de versie die in de meeste literatuur besproken wordt. Deze variant heeft weinig praktische waarde. Ze verduidelijkt echter aanzienlijk de analogie tussen het Viterbi-algoritme en het EstiHMM-algoritme en is in dat opzicht van theoretisch belang.

We tonen de relevantie van deze twee bijdragen aan door experimenten uit te voeren en een concrete toepassing te behandelen. We bespreken enkele interessante verbanden tussen beide algoritmes en onderzoeken welke meerwaarde het EstiHMM-algoritme biedt ten opzichte van het Viterbi-algoritme. We tonen aan de hand van concrete voorbeelden dat het EstiHMM-algoritme er in slaagt de eventuele robuustheidsproblemen van het Viterbi-algoritme op te vangen. In die gevallen waar het Viterbi-algoritme met robuustheidsproblemen kampt, zal het EstiHMM-algoritme meerdere oplossingen suggereren. Dit is een indicatie dat we over te weinig data beschikken om op betrouwbare wijze tot één enkele, precieze oplossing te komen.

Een eenvoudiger variant van het EstiHMM-algoritme, die – in tegenstelling tot deze masterproef – een minder algemeen geval behandelt, werd eerder dit jaar door mijn promotor, Gert de Cooman, en mezelf tot een artikel verwerkt. Dit artikel [1] werd ingestuurd voor presentatie op ISIPTA 2011, één van de toonaangevende conferenties op het gebied van imprecieze waarschijnlijkheden. Recent kregen we te horen dat ons artikel aanvaard is.

1.3 Opbouw van deze masterproef

We beginnen deze masterproef met het bespreken van enkele belangrijke begrippen en concepten waar we later vaak en intensief gebruik van zullen maken. Dit gedeelte neemt – met inbegrip van deze inleiding – de eerste vijf hoofdstukken in beslag.

In hoofdstuk 2 leggen we enkele elementaire concepten uit aan de hand van het eenvoudigste voorbeeld van een Markovmodel, namelijk een Markovketen. We bouwen hier vervolgens in hoofdstuk 3 op verder en voeren het uitgebreidere hidden Markovmodel in. We bespreken één van de belangrijkste algoritmes voor dit model, namelijk het Viterbi-algoritme en beschrijven hiervan een variant die algemener is dan de versie die in de meeste literatuur besproken wordt.

1. Estimation in imprecise Hidden Markov Models

In hoofdstuk 4 introduceren we de basisconcepten van de theorie van coherente onderp-revisies. Deze theorie was het belangrijkste wiskundige hulpmiddel om het EstiHMM-algoritme te ontwikkelen. Ze laat toe om vlot te rekenen met imprecieze waarschijnlijkheden en veralgemeent het gekende begrip verwachtingswaarde uit de precieze waarschijnlijkheidsleer.

Deze theorie wordt vervolgens in hoofdstuk 5 gebruikt om het hidden Markovmodel van hoofdstuk 3 uit te breiden zodat het in staat is om imprecisie te modelleren. We komen op die manier tot het imprecies hidden Markovmodel waarvoor het EstiHMM-algoritme ontwikkeld is.

Nu de meeste belangrijke begrippen zijn behandeld, zijn we klaar om het EstiHMM-algoritme te bespreken. Hoofdstuk 6 beschrijft eerst de algemene principes waarop dit algoritme gebaseerd is en geeft vervolgens aan hoe deze principes gebruikt kunnen worden om de maximale sequenties van een imprecies hidden Markovmodel efficiënt te bepalen. We bewijzen dat het algoritme een exacte oplossing geeft voor dit probleem en dat de complexiteit ervan, mits een goede implementatie, beperkt blijft.

In hoofdstuk 7 voeren we met dit algoritme enkele experimenten uit. We onderzoeken hoe de resultaten van het EstiHMM-algoritme evolueren naarmate de imprecisie van het gebruikte model toeneemt en vestigen de aandacht op enkele interessante verbanden met de resultaten van het Viterbi-algoritme. We sluiten het hoofdstuk af met een illustratie van de efficiëntie van het algoritme.

Tot slot geven we in hoofdstuk 8 een eenvoudige toepassing van het EstiHMM-algoritme. We gebruiken het algoritme om fouten die in een tekst zijn geslopen op automatische wijze te verbeteren. We vergelijken de resultaten van het Viterbi-algoritme en het EstiHMM-algoritme en tonen aan dat het EstiHMM-algoritme in staat is om eventuele robuustheidsproblemen van het Viterbi-algoritme op te vangen.

Deze masterproef eindigt in hoofdstuk 9 met een creatieve poging om de essentie van al het voorgaande beknopt te beschrijven. We noemen dit het besluit.

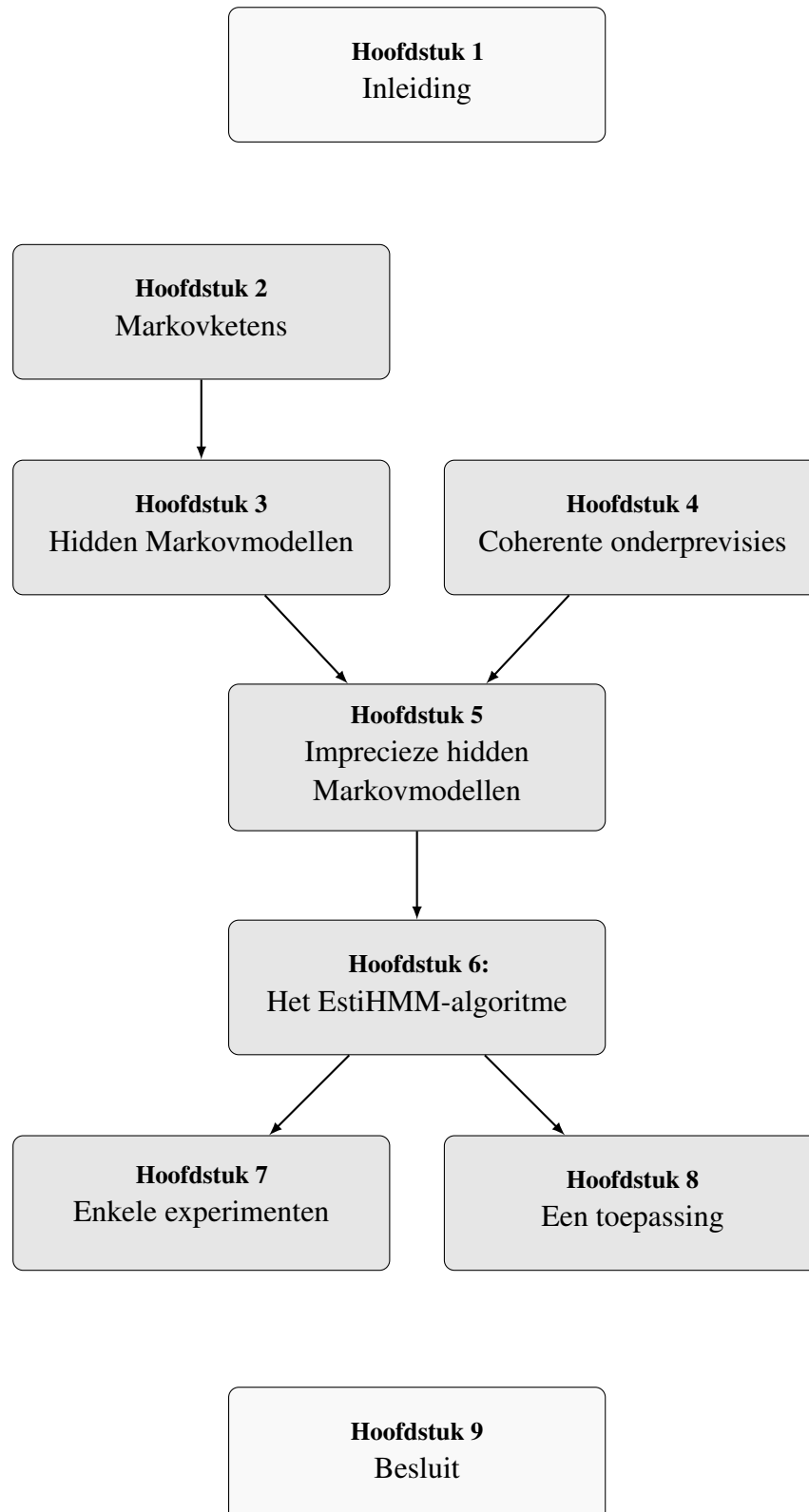
Achteraan dit werk zijn ook nog enkele bijlagen opgenomen.

Bijlage A bevat bewijzen voor enkele in deze masterproef gebruikte eigenschappen. Deze bewijzen zijn op zich minder belangrijk aangezien men ze ook kan terugvinden in de literatuur. In vele gevallen is dit het referentiewerk over imprecieze waarschijnlijkheid van Peter Walley [2]. Ze werden toegevoegd voor de volledigheid en om dit werk meer op zichzelf te laten staan. Meer belangrijke of zelf ontwikkelde bewijzen staan niet in bijlage A. Deze zijn terug te vinden in de hoofdtekst van dit werk.

In bijlage B zijn alle data verzameld die gebruikt werden om de toepassing in hoofdstuk 8 uit te werken. Op het einde van de bijlage vindt men de resultaten die met die data werden bekomen.

In bijlage C vindt men tot slot een implementatie van zowel het Viterbi-algoritme als het EstiHMM-algoritme. De code hiervan is geschreven in de programmeertaal Python.

We schematiseren de hier besproken opbouw van dit werk in het onderstaande overzicht.



1.4 Over de taal in dit werk

Het merendeel van de literatuur over imprecieze waarschijnlijkheid en andere onderwerpen die in deze masterproef aan bod komen is Engelstalig. Tijdens het schrijven van dit werk werd ik dan ook soms geconfronteerd met het gebrek aan een eenduidige vertaling van de gebruikte terminologie. In vele gevallen werd reeds door andere mensen een – vaak geslaagde – poging gedaan om een equivalente Nederlandstalige term in te voeren. In de meeste gevallen neem ik deze term met graagte over. Er zijn echter ook woorden waarvoor ik de Engelse term prefereer omdat deze naar mijn gevoel een op zichzelf staand begrip geworden zijn, dat losstaat van de gebruikte taal. Het opvallendste voorbeeld is waarschijnlijk het hidden Markovmodel. Ik verkies deze term boven het volledig Nederlandstalige equivalent, zijnde een verborgen Markovmodel. Ik geef toe dat deze keuze voor discussie vatbaar is en wil zeker niet pretenderen dat mijn keuze op meer gebaseerd is dan een soort van buikgevoel.

Tot slot wil ik nog het volgende kwijt: spelling is niet mijn sterkste punt. De kans is dus groot dat in dit werk hier en daar nog een dt-fout word gemaakt. Ik kan hier mee leven. Het hoofddoel van taal is wat mij betreft communicatie. Als dit werk er in slaagt mijn ideeën en bevindingen duidelijk over te brengen, dan zal ik een tevreden mens zijn. Wie weet vindt u het zelfs boeiend. Ik hoop alvast van wel.

Hoofdstuk 2

Markovketens

“La théorie des probabilités n’est, au fond, que le bon sens réduit au calcul.”

– Pierre-Simon Laplace

2.1 Inleiding

Heel veel van de dingen die we dagelijks om ons heen waarnemen bevatten een zekere logica. Wie over een gezonde hoeveelheid boerenverstand beschikt, herkent al vlug allerlei eenvoudige verbanden en patronen in soms zeer complexe zaken. Weermannen breken zich dagelijks het hoofd om het weer te voorspellen en toch zal een klein kind dat na twee dagen zonnig weer vraagt om morgen naar zee te gaan, waarschijnlijk gelijk hebben als het denkt dat morgen het zonnetje ook wel zal schijnen. Hoewel het weer van zeer veel vaak ongekende factoren afhangt – wie heeft nog niet gehoord dat een al te enthousiaste vlinder in het Braziliaanse regenwoud voor die plensbui in Texas verantwoordelijk was? – zijn er blijkbaar toch enkele eenvoudige verbanden die iedereen spontaan door heeft. Door dergelijke eenvoudige verbanden om te zetten in een wiskundig model, kunnen we met behulp van de wiskunde ons gezond verstand een handje helpen om problemen die te moeilijk worden voor het menselijk brein, toch verrassend snel en elegant op te lossen. Markovmodellen zijn hiervan een elementair, maar krachtig en populair voorbeeld.

Er bestaan verschillende soorten Markovmodellen. Twee van de meest belangrijke zijn de Markovketen [3] en het hidden Markovmodel [4]. Hoewel dit laatste model een belangrijke plaats inneemt in deze masterproef en dus verderop in detail zal behandeld worden, lijkt het ons nuttig om eerst wat nader in te gaan op de Markovketen. Dit model is eenvoudiger, en daarom meer geschikt om enkele basisideeën in te voeren die later ook zullen terugkomen in het hidden Markovmodel. We gebruiken het voorbeeld van weersvoorspelling om deze elementaire begrippen bevattelijk aan te brengen.

2.2 Toestandsveranderlijken

Het weer op een bepaalde dag kan beschouwd worden als een toevallige veranderlijke X die we de *toestandsveranderlijke* noemen. Deze veranderlijke kan verschillende waarden, *toestanden* genaamd, aannemen. Als we de werkelijkheid vereenvoudigen kunnen we bijvoorbeeld stellen dat het weer steeds zonnig, bewolkt of regenachtig is. De verzameling van de mogelijke toestanden noteren we als \mathcal{X} . Voor ons voorbeeld geldt dat

$$\mathcal{X} = \{\text{zonnig}, \text{bewolkt}, \text{regen}\}.$$

Deze *toestandsverzameling* moet aan enkele vereisten voldoen. Enerzijds dient de verzameling *uitputtend* te zijn, wat wil zeggen dat er steeds één van de mogelijke toestanden optreedt. In ons vereenvoudigd model gaan we er dus van uit dat er niet nog een vierde toestand *sneeuw* bestaat. Anderzijds moeten de toestanden elkaar wederzijds uitsluiten. Hiermee bedoelen we dat er slechts één toestand tegelijk kan optreden en dat het bijvoorbeeld niet tegelijk kan regenen en bewolkt zijn. Natuurlijk zal het in de werkelijkheid altijd bewolkt zijn als het regent, maar dat bedoelen we niet in ons model. Bewolkt wil in het model zeggen dat het een grijze dag is, maar dat het niet regent.

In het kader van deze masterproef maken we voor de gebruikte toestandsruimtes ook nog een bijkomende veronderstelling.

Aanname 1

Alle in deze masterproef beschouwde toestandsverzamelingen zijn eindig, ze bezitten met andere woorden slechts een eindig aantal elementen. Dit geldt ook voor de verzamelingen van mogelijke observaties die we later in paragraaf 3.2 zullen invoeren.

De verzameling \mathbb{N} van de natuurlijke getallen laten we bijvoorbeeld niet toe als toestandsverzameling. Een rechtstreeks gevolg hiervan is dat alle functies die we op deze eindige verzamelingen definiëren begrensd zijn. Deze beperking wordt in de literatuur zowel voor theoretische als praktische problemen zeer vaak opgelegd, en doet in die zin dus niet af aan de bruikbaarheid van de in deze masterproef gepresenteerde resultaten.

De verschillende toestanden in \mathcal{X} zullen we algemeen noteren als x . Zo wordt met $X = x$ bedoeld dat de veranderlijke X de waarde x aanneemt, waarbij x bijvoorbeeld de toestand *zonnig* is.

Het weer gedurende een bepaalde periode is dan een sequentie $X_{1:n} = (X_1, X_2, \dots, X_n)$ van veranderlijken, waarbij n het aantal opeenvolgende dagen is dat we beschouwen. Als we het weer gedurende één week beschouwen, dan geldt bijvoorbeeld dat $n = 7$. De veranderlijken X_k nemen hierbij voor alle $k \in \{1, \dots, n\}$ waarden aan in \mathcal{X}_k . We kunnen ook zeggen dat de sequentie $X_{1:n}$ waarden $x_{1:n}$ aanneemt in $\mathcal{X}_{1:n} = \times_{i=1}^n \mathcal{X}_i$. De uitdrukking $X_{1:n} = x_{1:n}$ is equivalent met zeggen dat $X_1 = x_1, \dots, X_n = x_n$.

2.3 Een model op basis van waarschijnlijkheden

2.3.1 Lokale modellen

We zullen nu onze kennis over het weer op een bepaalde dag beschrijven met behulp van een waarschijnlijkheidsmodel. We kunnen dit bijvoorbeeld doen door een massafunctie $q_1(X_1)$ te beschouwen, gedefinieerd op \mathcal{X}_1 . Een dergelijke massafunctie geeft voor elke mogelijk waarde van de toestandsveranderlijke X_1 de waarschijnlijkheid dat deze waarde optreedt. De waarschijnlijkheid dat $X_1 = x_1$ noteren we als $q_1(x_1)$. Voor België zouden we de onderstaande waarden kunnen gebruiken.

$$\begin{aligned} q_1(\text{zonnig}) &= 0,15 = 15\% \\ q_1(\text{bewolkt}) &= 0,35 = 35\% \\ q_1(\text{regen}) &= 0,5 = 50\% \end{aligned}$$

De verzameling van alle mogelijke op \mathcal{X}_1 gedefinieerde massafuncties zullen we noteren als $\Sigma_{\mathcal{X}_1}$.

$$q \in \Sigma_{\mathcal{X}_1} \Leftrightarrow \sum_{x \in \mathcal{X}_1} q(x) = 1 \text{ en } \forall x \in \mathcal{X}_1 : q(x) \geq 0 \quad (2.1)$$

Hierbij impliceert de voorwaarde $\sum_{x \in \mathcal{X}_1} p(x) = 1$ dat de kans dat één van de mogelijke toestanden optreedt 100% is, en volgt uit de voorwaarde $p(x) \geq 0$ dat een kans nooit negatief kan zijn.

Als we vervolgens een waarschijnlijkheidsmodel opstellen voor het weer op dag twee, dan zouden we bijvoorbeeld dezelfde massafunctie kunnen gebruiken als voor de eerste dag. Hiermee negeren we dan echter het verband tussen het weer van vandaag en dat van gisteren. Op een willekeurige dag is de kans op zonnig weer in ons voorbeeld gelijk aan 15%, maar als we weten dat het gisteren zonnig was, dan zal de kans dat het vandaag ook zonnig is behoorlijk groter zijn dan 15%, eerder bijvoorbeeld 70%. Dergelijke afhankelijkheden kunnen we uitdrukken met behulp van een conditioneel waarschijnlijkheidsmodel. Voor de toestand X_2 van het weer op dag twee beschouwen we de massafunctie $q_2(X_2|X_1)$, gedefinieerd op \mathcal{X}_2 en conditioneel op X_1 . Voor elke $x_1 \in \mathcal{X}_1$ hebben we dus een verschillend waarschijnlijkheidsmodel $q_2(X_2|x_1)$. De uitspraak hierboven over de kans op zonnig weer op dag twee als het op dag één ook zonnig was kunnen we dan wiskundig uitdrukken als $q_2(\text{zonnig}|x_1)$, waarbij in dit geval $x_1 = \text{zonnig}$.

Het model voor het weer op dag drie hangt op zijn beurt af van het weer gedurende de eerste twee dagen. We hebben in dit geval een massafunctie $q_3(X_3|X_{1:2})$ voor de toestand X_3 , gedefinieerd op \mathcal{X}_3 en conditioneel op $X_{1:2}$. Op analoge wijze kunnen we de modellen voor de volgende dagen definiëren, waarbij we tot slot voor de toestand X_n op de laatste dag het model $q_n(X_n|X_{1:n-1})$ beschouwen, gedefinieerd op \mathcal{X}_n en conditioneel op $X_{1:n-1}$.

2.3.2 Een globaal model

We willen nu uitgaande van de verschillende lokale modellen die we hierboven geïntroduceerd hebben een globaal model opbouwen dat ons iets zegt over de toestand van een hele *sequentie*, in plaats van enkel over de toestand van één enkel element. We willen bijvoorbeeld uitspraken kunnen doen over het weer gedurende een hele week, en niet enkel over de afzonderlijke dagen in die week. Een mogelijke – en voor velen vermoedelijk belangrijke – vraag die we kunnen stellen is wat de waarschijnlijkheid is dat het de hele week zonnig zal zijn. Wiskundig gaan we dus op zoek naar het globaal model $p_{1:n}(X_{1:n})$, dat ons voor elke mogelijke waarde $x_{1:n} \in \mathcal{X}_{1:n}$ van de sequentie $X_{1:n}$ de waarschijnlijkheid geeft dat $X_{1:n} = x_{1:n}$. Als $x_k = \text{zonnig}$ voor alle $k \in \{1, \dots, 7\}$, dan is $p_{1:7}(x_{1:7})$ de waarschijnlijkheid dat het de hele week zonnig zal zijn.

We lossen dit probleem recursief op. Dit wil zeggen dat we eerst een klein deelprobleem oplossen, en vervolgens een steeds groter gedeelte van het probleem oplossen, waarbij we gebruik maken van de oplossingen die we eerder reeds bekomen hebben voor de deelproblemen.

Het kleinste deelprobleem hebben we eigenlijk reeds opgelost, namelijk wat is de kans dat het op de eerste dag zonnig weer zal zijn. Deze kans is gelijk aan $q_1(x_1)$. Vervolgens gaan we op zoek naar de kans $p_{1:2}(x_{1:2})$ dat het zowel op de eerste als de tweede dag zonnig is. Dit kunnen we doen door de kans dat het op dag één zonnig is te vermenigvuldigen met de kans dat het op de tweede dag zonnig is, *conditioneel* op het feit dat we weten dat het de eerste dag al zonnig was. In formules komt dit overeen met $p_{1:2}(x_{1:2}) = q_1(x_1)q_2(x_2|x_1)$. Op analoge wijze vinden we dat $p_{1:3}(x_{1:3}) = p_{1:2}(x_{1:2})q_3(x_3|x_{1:2}) = q_1(x_1)q_2(x_2|x_1)q_3(x_3|x_{1:2})$. Tot slot vinden we zo ook $p_{1:n}(x_{1:n})$, wat de kans is dat $X_{1:n} = x_{1:n}$.

$$\begin{aligned} p_{1:n}(x_{1:n}) &= p_{1:n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1}) \\ &= p_{1:n-2}(x_{1:n-2})q_{n-1}(x_{n-1}|x_{1:n-2})q_n(x_n|x_{1:n-1}) \\ &= \dots \\ &= q_1(x_1)q_2(x_2|x_1) \cdots q_{n-1}(x_{n-1}|x_{1:n-2})q_n(x_n|x_{1:n-1}) \\ &= q_1(x_1) \prod_{i=2}^n q_i(x_i|x_{1:i-1}). \end{aligned}$$

Dit is geldig voor elke mogelijke sequentie $x_{1:n} \in \mathcal{X}_{1:n}$, en dus vinden we onderstaande uitdrukking voor ons globaal model.

$$p_{1:n}(X_{1:n}) = q_1(X_1) \prod_{i=2}^n q_i(X_i|X_{1:i-1}).$$

Deze uitdrukking werd recursief opgebouwd van links naar rechts. Hiermee bedoel ik dat we begonnen zijn met een model voor de eerste veranderlijke, dat we hieruit een model opgebouwd hebben voor de eerste twee veranderlijken samen, en dat we zo het model steeds verder uitgebreid hebben. Men kan echter ook omgekeerd te werk gaan,

namelijk van rechts naar links. We zullen in latere hoofdstukken zien dat deze laatste manier van werken zeer nuttig kan zijn, en we zullen ze daarom vanaf nu consequent toepassen.

We voeren voor alle $k \in \{2, \dots, n\}$ de conditionele globale modellen $p_k(X_{k:n}|X_{1:k-1})$ in.¹ Deze modellen geven voor elke $x_{k:n} \in \mathcal{X}_{k:n}$ de waarschijnlijkheid dat $X_{k:n} = x_{k:n}$, en dit conditioneel op de waarde $x_{1:k-1}$ van de sequentie $X_{1:k-1}$. De recursieve uitdrukkingen voor deze modellen worden gegeven door

$$p_k(X_{k:n}|X_{1:k-1}) = \begin{cases} q_n(X_n|X_{1:n-1}) & \text{als } k = n \\ q_k(X_k|X_{1:k-1})p_{k+1}(X_{k+1:n}|X_{1:k}) & \text{als } k = 2, \dots, n-1. \end{cases} \quad (2.2)$$

We voeren ook nog het globaal model $p_1(X_{1:n})$ in. Dit geeft voor elke $x_{1:n} \in \mathcal{X}_{1:n}$ de waarschijnlijkheid dat $X_{1:n} = x_{1:n}$, en is dus eigenlijk gelijk aan het eerder ingevoerde model $p_{1:n}(X_{1:n})$. We kiezen er bewust voor om vanaf nu een andere notatie te gebruiken omdat we zo de aandacht willen vestigen op de analogie met de conditionele modellen $p_k(X_{k:n}|X_{1:k-1})$ en willen duidelijk maken dat het model wordt opgebouwd door recursie van rechts naar links. De recursiebetrekking voor dit globaal model is

$$p_1(X_{1:n}) = q_1(X_1)p_2(X_{2:n}|X_1). \quad (2.3)$$

We kunnen de recursiebetrekkingen (2.2) en (2.3) steeds opnieuw toepassen tot we uiteindelijk vinden dat

$$p_1(X_{1:n}) = q_1(X_1) \prod_{i=2}^n q_i(X_i|X_{1:i-1}), \quad (2.4)$$

en dat voor alle $k \in \{2, \dots, n\}$

$$p_k(X_{k:n}|X_{1:k-1}) = \prod_{i=k}^n q_i(X_i|X_{1:i-1}). \quad (2.5)$$

De uitdrukking in het rechterlid van vergelijking (2.5) is natuurlijk dezelfde als diegene die we eerder afgeleid hebben toen we recursie in de andere richting hebben toegepast.

2.4 De Markovvoorwaarde

Een probleem met het hierboven beschreven model is dat het lokale conditionele waarschijnlijkheidsmodel van een veranderlijke afhankelijk is van alle voorgaande veranderlijken. Het weer op het einde van de week is afhankelijk van hoe het weer was gedurende alle vorige dagen. Hoewel dit in de realiteit natuurlijk het geval is, is het

1. We zouden net zo goed de notatie $p_{k:n}(X_{k:n}|X_{1:k-1})$ kunnen gebruiken. We verkiezen echter de variant $p_k(X_{k:n}|X_{1:k-1})$ omdat deze de notatie minder overlaadt.

belangrijk te beseffen dat dit leidt tot vrij complexe berekeningen. Het is daarom vaak nuttig om te veronderstellen dat een lokaal waarschijnlijkheidsmodel énkél afhangt van de toestand van de vorige veranderlijke. Onze vermoedens over het weer van morgen, hangen dus enkel af van het weer van vandaag, en niet van dit van gisteren. We noemen deze onafhankelijkheidsvoorwaarde de *Markovvoorwaarde* en spreken in dat geval van een *Markovmodel*, waarvan de in dit hoofdstuk besproken Markovketen een eenvoudig voorbeeld is. De Markovvoorwaarde is een sterke vereenvoudiging die kan leiden tot minder nauwkeurige resultaten, maar wel als groot voordeel heeft dat de berekeningen heel wat eenvoudiger worden. Voor vele toepassingen blijkt dit model goede resultaten te geven, en het wordt dan ook wereldwijd in diverse domeinen met succes toegepast.

Uit deze Markovvoorwaarde volgt voor onze Markovketen onmiddellijk dat voor alle $k \in \{2, \dots, n\}$ geldt dat $q_k(X_k | X_{1:k-1}) = q_k(X_k | X_{k-1})$. De uitdrukking (2.4) voor het globaal model wordt hierdoor vereenvoudigd tot

$$p_1(X_{1:n}) = q_1(X_1) \prod_{i=2}^n q_i(X_i | X_{i-1}). \quad (2.6)$$

Ook uitdrukking (2.5) kan vereenvoudigd worden. We vinden voor alle $k \in \{2, \dots, n\}$ dat

$$p_k(X_{k:n} | X_{k-1}) = \prod_{i=k}^n q_i(X_i | X_{i-1}). \quad (2.7)$$

We geven tot slot nog de door de Markovvoorwaarde vereenvoudigde versies van de recursieve uitdrukkingen (2.2) en (2.3) waaruit deze modellen werden afgeleid. Merk op dat vergelijking (2.9) identiek is aan vergelijking (2.3) en hier dus enkel voor de volledigheid wordt vermeld.

$$p_k(X_{k:n} | X_{k-1}) = \begin{cases} q_n(X_n | X_{n-1}) & \text{als } k = n \\ q_k(X_k | X_{k-1}) p_{k+1}(X_{k+1:n} | X_k) & \text{als } k = 2, \dots, n-1 \end{cases} \quad (2.8)$$

$$p_1(X_{1:n}) = q_1(X_1) p_2(X_{2:n} | X_1). \quad (2.9)$$

2.5 Stationariteit

Vaak legt men naast de Markovvoorwaarde nog een extra vereenvoudiging op, namelijk stationariteit. Men spreekt van een *stationaire* Markovketen als de gebruikte lokale conditionele modellen niet afhangen van de positie in de keten. Men bedoelt hiermee dat $\mathcal{X}_k = \mathcal{X}$ voor alle $k \in \{1, \dots, n\}$ en dat voor alle $x \in \mathcal{X}$, alle $i \in \{2, \dots, n\}$ en alle $j \in \{2, \dots, n\}$ geldt dat $q_i(\cdot | x) = q_j(\cdot | x)$. De eigenschappen en algoritmen die verder ter sprake zullen komen zullen deze stationariteit – tenzij expliciet anders vermeld – niet vereisen, maar voor praktische toepassingen neemt men wel vaak stationariteit aan, aangezien dit het aantal te bepalen parameters aanzienlijk vermindert.

2.6 Een model op basis van verwachtingswaarden

2.6.1 Lokale modellen

Tot nu toe hebben we ons model steeds uitgedrukt met waarschijnlijkheden. Dit is logisch, aangezien dit de eenvoudigste en meest bevattelijke vorm is waarmee we het principe van een Markovketen kunnen illustreren. Voor het vervolg van deze uiteenzetting zal het echter nuttig zijn om ditzelfde model uit te drukken met verwachtingswaarden. Deze voorstelling is volkomen equivalent, maar zal ons later in hoofdstuk 5 in staat stellen om op een elegante en efficiënte manier over te gaan van een precies naar een imprecies Markovmodel.

Met het lokale waarschijnlijkheidsmodel $q_1(X_1)$ correspondeert een lineaire verwachtingsfunctionaal – of *previsie* – Q_1 , gedefinieerd op de verzameling $\mathcal{L}(\mathcal{X}_1)$ van alle reëelwaardige afbeeldingen – of functies – f op \mathcal{X}_1 . Dit *marginaal model* wordt gegeven door

$$Q_1(f) = \sum_{z_1 \in \mathcal{X}_1} f(z_1)q_1(z_1).$$

Op analoge wijze komt voor alle $k \in \{2, \dots, n\}$ met $q_k(X_k|X_{k-1})$ een *transitiemodel* overeen dat gegeven wordt door

$$Q_k(f|X_{k-1}) = \sum_{z_k \in \mathcal{X}_k} f(z_k)q_k(z_k|X_{k-1}).$$

Voor het gemak van notatie zullen we soms naar het marginaal model Q_1 verwijzen als zijnde een conditionele previsie $Q_1(\cdot|X_0)$. Hierbij is X_0 een veranderlijke die slechts één enkele waarde kan aannemen en waarvan de waarde dus zeker is. Conditioneren op zo'n veranderlijke heeft logischerwijs geen enkel effect. Door dit notationeel trucje hoeven we eigenschappen die zowel geldig zijn voor het marginaal model als voor de transitie modellen geen twee maal te vermelden of te bewijzen. Het volstaat om dit enkel te doen voor de conditionele transitie modellen, aangezien we het marginaal model kunnen beschouwen als een speciaal soort van transitie model. Volkomen analoog zullen we ook soms naar de waarschijnlijkheidsverdelingen $q_1(X_1)$ en $p_1(X_{1:n})$ verwijzen als zijnde de conditionele verdelingen $q_1(X_1|X_0)$ en $p_1(X_{1:n}|X_0)$.

De equivalentie tussen een model met previsions en een model met waarschijnlijkheden kan als volgt ingezien worden. Als we beschikken over een massafunctie, dan is de overeenkomstige previsie volledig bepaald door de hierboven ingevoerde definities. Als we echter over een previsie beschikken, dan bepaalt deze ook op unieke wijze de overeenstemmende massafunctie.

Dit laatste kan aangetoond worden door een speciaal soort functies te beschouwen, namelijk indicatoren. Een indicator is een functie die correspondeert met een bepaalde gebeurtenis. Als deze gebeurtenis optreedt, dan heeft de indicator de waarde één, zo niet, dan heeft de indicator de waarde nul. Het meest eenvoudige type gebeurtenis is het optreden van een bepaalde waarde x_k voor de veranderlijke X_k . De indicator die met

deze gebeurtenis $\{x_k\}$ overeenstemt² wordt gegeven door

$$\mathbb{I}_{\{x_k\}}(X_k) = \begin{cases} 1 & \text{als } X_k = x_k \\ 0 & \text{als } X_k \neq x_k. \end{cases}$$

Door deze indicator in te vullen in de lokale previsies, vinden we de gevraagde equivalentie tussen een previsie en zijn corresponderend waarschijnlijkheidsmodel. Voor alle $k \in \{1, \dots, n\}$ geldt namelijk dat

$$\begin{aligned} Q_k(\mathbb{I}_{\{x_k\}} | X_{k-1}) &= \sum_{z_k \in \mathcal{X}_k} \mathbb{I}_{\{x_k\}}(z_k) q_k(z_k | X_{k-1}) \\ &= q_k(x_k | X_{k-1}). \end{aligned} \quad (2.10)$$

Merk op dat dit meteen ook het nut van de notatie $Q_1(\cdot | X_0)$ als alternatief voor $Q_1(\cdot)$ illustreert. Door deze notatie te gebruiken hoeven we geen afzonderlijke versie van vergelijking (2.10) te voorzien voor de previsie $Q_1(\cdot)$.

Tot slot voeren we, om de notatie te verlichten, voor $Q_k(\mathbb{I}_{\{x_k\}} | X_{k-1})$ nog de alternatieve notatie $Q_k(\{x_k\} | X_{k-1})$ in. Hierbij identificeren we de indicator $\mathbb{I}_{\{x_k\}}$ dus met zijn corresponderende gebeurtenis $\{x_k\}$. Ook voor andere indicatoren zullen we deze identificatie in het vervolg van deze masterproef soms toepassen.

2.6.2 Een globaal model

Aan de hand van de hierboven ingevoerde lokale previsies willen we nu een globaal model opbouwen met verwachtingswaarden. Dit globaal model wordt voor alle functies $f \in \mathcal{L}(\mathcal{X}_{1:n})$ gedefinieerd door

$$P_1(f) = \sum_{z_{1:n} \in \mathcal{X}_{1:n}} f(z_{1:n}) p_1(z_{1:n}). \quad (2.11)$$

We voeren ook de conditionele previsies

$$P_k(f | X_{k-1}) = \sum_{z_{k:n} \in \mathcal{X}_{k:n}} f(z_{k:n}) p_n(z_{k:n} | X_{k-1}) \quad (2.12)$$

in, die gedefinieerd zijn voor alle $f \in \mathcal{L}(\mathcal{X}_{k:n})$ en alle $k \in \{2, \dots, n\}$.

Met hetzelfde notationele trucje dat we in paragraaf 2.6.1 al introduceerden zullen we –waar nuttig– naar $P_1(\cdot)$ verwijzen als de conditionele previsie $P_1(\cdot | X_0)$. Hierdoor wordt vergelijking (2.12) ook zinvol voor $k = 1$, en wordt deze in dat geval identiek aan vergelijking (2.11).

2. Een gebeurtenis wordt wiskundig uitgedrukt als een deelverzameling. Als we bijvoorbeeld met een dobbelsteen gooien, dan komt de gebeurtenis “we gooien een even getal” overeen met de deelverzameling $\{2, 4, 6\}$ van de verzameling van $\{1, 2, 3, 4, 5, 6\}$ van de mogelijke uitkomsten. De gebeurtenis “we gooien een zes” komt overeen met de deelverzameling $\{6\}$. Dit verklaart meteen de door ons gebruikte notatie $\{x_k\}$ voor de gebeurtenis $X_k = x_k$.

Al deze modellen kunnen recursief opgebouwd worden. Er geldt namelijk voor alle $k \in \{1, \dots, n-1\}$ en alle $f \in \mathcal{L}(\mathcal{X}_{k:n})$ dat

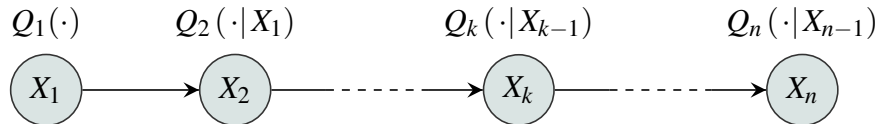
$$\begin{aligned} P_k(f|X_{k-1}) &= \sum_{z_{k:n}} f(z_{k:n}) q_k(z_k|X_{k-1}) p_{k+1}(z_{k+1:n}|z_k) \\ &= \sum_{z_k} q_k(z_k|X_{k-1}) \left(\sum_{z_{k+1:n}} f(z_{k:n}) p_{k+1}(z_{k+1:n}|z_k) \right) \\ &= \sum_{z_k} q_k(z_k|X_{k-1}) (P_{k+1}(f(z_k, X_{k+1:n})|z_k)) \\ &= Q_k(P_{k+1}(f(X_k, X_{k+1:n})|X_k) | X_{k-1}). \end{aligned}$$

De gezochte recursieve betrekkingen zijn dus

$$P_k(f|X_{k-1}) = \begin{cases} Q_n(f|X_{n-1}) & \text{als } k = n \\ Q_k(P_{k+1}(f(X_k, X_{k+1:n})|X_k) | X_{k-1}) & \text{als } k = 1, \dots, n-1. \end{cases} \quad (2.13)$$

2.7 Grafische voorstelling

Een Markovketen kan bevattelijk voorgesteld worden door het onderstaande grafische model [5].



Het is dus een sequentie van toevallige veranderlijken, *toestandsveranderlijken* genaamd, die elk beschreven worden aan de hand van een lokaal model op basis van verwachtingswaarden, een *previsie* genaamd. Voor de eerste toestand is dit een *marginaal* model. Voor de overige toestanden is dit een *transitiemodel*, conditioneel op de waarde van de vorige toestand. Uit deze lokale modellen kan vervolgens een globaal model worden opgebouwd, zoals uiteengezet in paragraaf 2.6.2.

De pijlen die in deze grafische voorstelling geïntroduceerd werden geven een richting aan ons model. Voor een Markovketen is deze richting voor de hand liggend, aangezien ze reeds aangegeven wordt door de nummering van de veranderlijken. Voor complexere modellen, zoals bijvoorbeeld het hidden Markovmodel, zal deze nummering echter niet langer volstaan, en zullen de pijlen de onderlinge verbanden tussen de verschillende veranderlijken duidelijk maken. Ze laten ook toe om de begrippen *moeder* en *kind* te definiëren. De moeder³ is de veranderlijke die een andere veranderlijke, het kind

3. Men kan terecht opmerken dat we hier net zo goed over de vader zouden kunnen spreken. Van zodra we echter uitspraken doen zoals “Als de vader/moeder gekend is...”, lijkt het me in de huidige maatschappelijke context veiliger om het over de moeder te hebben.

genaamd, direct voorafgaat. Elke pijl verbindt dus telkens een moeder met haar kind. De Markovvoorwaarde impliceert vervolgens dat conditioneel op de moeder, het model voor de toestand van het kind onafhankelijk is van al zijn voorouders [5]. De voorouders van een kind zijn daarbij al de veranderlijken die zijn moeder voorafgaan in het model.

Het is belangrijk om te beseffen dat deze conditionele onafhankelijkheid enkel geldt als de waarde van de moeder echt gekend is. Zolang die waarde niet vastligt, is er wel degelijk een invloed van de voorouders, omdat deze de waarde van de moeder beïnvloeden en op die manier dus ook die van het kind.

Een andere manier om de onafhankelijkheidsrelaties te beschrijven, is zeggen dat conditioneel op zijn moeder, een veranderlijke enkel afhankelijk is van zijn moeder en van zijn eigen nakomelingen. De nakomelingen van een veranderlijke zijn hierbij alle veranderlijken die volgens de richting die door de pijlen wordt aangegeven na deze veranderlijke komen.

Hidden Markovmodellen

“What if what seems to be isn’t true and what is true is behind what seems to be?”

– anonieme bedenking na het bekijken van *The Matrix*

3.1 Inleiding

Volmaaktheid is niet van deze wereld, al wordt daar wel vaak impliciet van uit gegaan. In het vorige hoofdstuk over Markovketens hebben we bijvoorbeeld verondersteld dat we de onderzochte toestanden – of dus het weer gedurende verschillende dagen – perfect konden waarnemen. Natuurlijk zijn we meestal in staat om te zien welk weer het is, maar laat ons om educatieve redenen even de volgende fictieve situatie beschouwen.

Stel u een man voor in zijn ivoren toren. In tegenstelling tot wat men in de sprookjes beweert, beschikken die lang niet altijd over de nodige vensters om de wereld te kunnen overschouwen. Deze man brengt hele dagen door in zijn ivoren toren, zonder enige drang om de buitenwereld te gaan opzoeken, en door zijn gebrek aan vensters is hij dus ook niet in staat om te zien welk weer het buiten is.

Op een dag beslist de man dat het tijd is voor verandering. Hij besluit af te dalen uit zijn toren en de wereld te gaan groeten, maar dat brengt natuurlijk grote problemen met zich mee: hij heeft geen flauw idee welk weer het buiten is. Men wil op de eerste dag dat men de wereld groet liever niet met een zonnebril naar buiten wandelen als de regen met bakken uit de lucht blijkt te vallen. Zijn vraag is nu dus hoe hij iets kan te weten komen over het weer buiten, zonder dat hij het ooit rechtstreeks heeft kunnen waarnemen.

Wat onze man in de ivoren toren wel weet, is dat er dagen zijn dat de deur van zijn kamer klemt, en dagen waarop dit helemaal niet het geval is. Bovendien weet hij dat er een verband bestaat tussen het weer en het klemmen van zijn deur. Als het buiten zonnig is, dan zal zijn deur waarschijnlijk niet klemmen, maar zeker is hij daar niet van. Op dezelfde manier zal zijn deur als het regent waarschijnlijk wél klemmen, maar het zou ook kunnen van niet, omdat het de dagen ervoor bijvoorbeeld heel erg droog

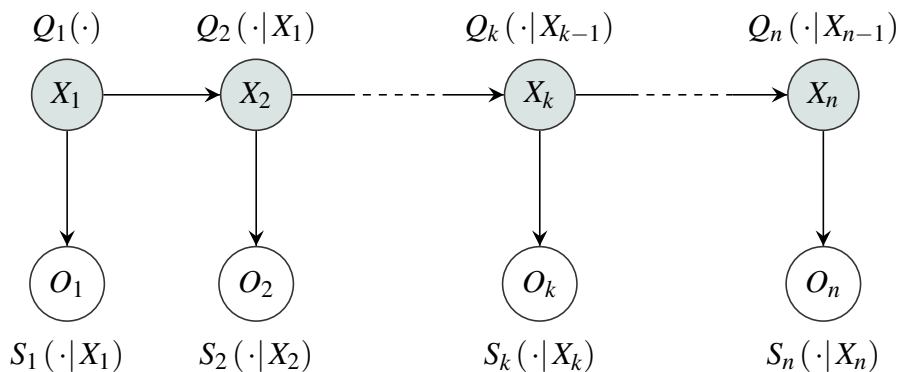
geweest is. Is het tot slot een bewolkte dag, dan weet hij eigenlijk niet goed wat het effect is op zijn deur, maar waarschijnlijk is er iets meer kans dat ze welk klemt dan dat ze niet klemt.

Er is dus duidelijk een verband tussen het weer en het klemmen van de deur. Hoewel hij door het bestuderen van de deur nooit met zekerheid zal kunnen zeggen wat het weer is, kan hij wel bepaalde vermoedens hebben. Als zijn deur vijf dagen op een rij niet klemt, en de laatste twee dagen wel, dan was het *waarschijnlijk* de eerste vijf dagen zonnig, werd het dan een dag bewolkt en regende het de laatste dag. Of zou het die laatste dag toch nog steeds bewolkt geweest zijn? U begrijpt dat dit geen voor de hand liggend probleem is, en de man in zijn ivoren toren zit dan ook met de handen in het haar. Gelukkig zal u hem op het einde van dit hoofdstuk kunnen helpen.

3.2 Observaties en verborgen toestanden

De manier waarop we dingen zullen te weten komen over verborgen toestanden – het weer buiten – die we niet kunnen waarnemen, is door verbanden te leggen tussen deze toestandsveranderlijken en andere elementen – het al dan niet klemmen van de deur – die we wel kunnen waarnemen. We noemen deze waarneembare elementen *observaties*. Een observatie wordt aangeduid door een veranderlijke O die waarden kan aannemen in de verzameling \mathcal{O} . Deze verzameling van mogelijke observaties voldoet aan dezelfde voorwaarden die we in paragraaf 2.2 hebben opgelegd aan de toestandsverzameling, namelijk dat de verzameling uitputtend en eindig moet zijn en dat de verschillende mogelijke observaties elkaar wederzijds moeten uitsluiten. De verschillende elementen van \mathcal{O} noteren we algemeen als o . Met $O = o$ bedoelen we dat de veranderlijke O de waarde o aanneemt.

De onderlinge verbanden tussen de toestandsveranderlijken en de observaties kunnen gemodelleerd worden met een hidden Markovmodel (HMM), dat schematisch is weergegeven in de onderstaande figuur.



Dit model lijkt sterk op een Markovketen, met dit verschil dat er behalve van een toestandssequentie, ook sprake is van een sequentie van observaties. Voor elke observatie

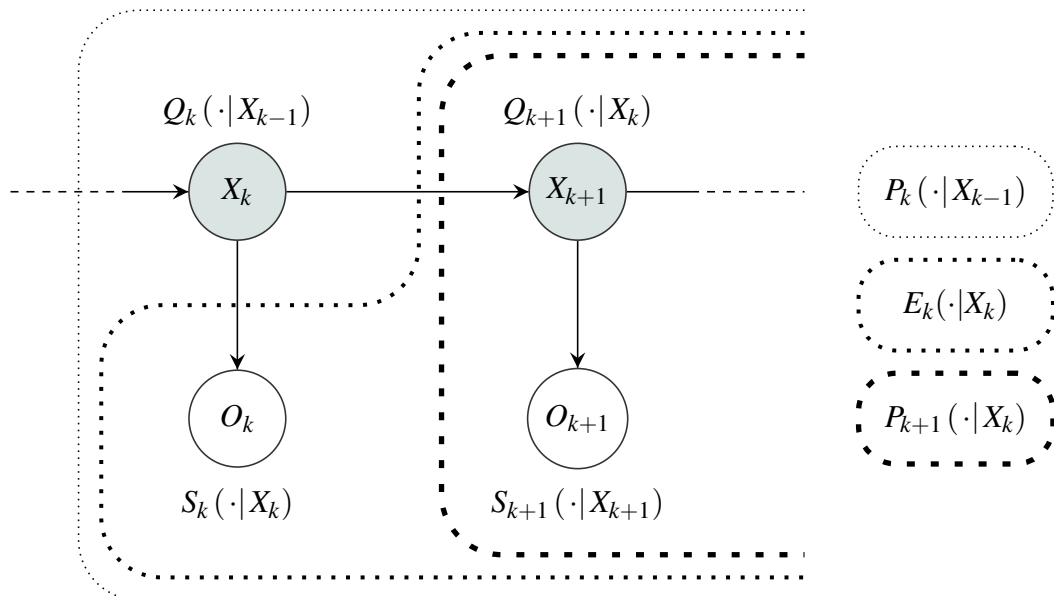
O_k beschikken we over een conditionele lokale previsie $S_k(\cdot|X_k)$, gedefinieerd op de verzameling $\mathcal{L}(\mathcal{O}_k)$ van alle reëelwaardige functies van O_k .

De pijlen introduceren opnieuw een richting in het model en definiëren zo de onafhankelijkheidsrelaties tussen de verschillende veranderlijken. Doordat het hidden Markovmodel in tegenstelling tot de Markovketen vertakkingen vertoont, volstaat het hier niet langer om te zeggen dat een veranderlijke, conditioneel op haar moeder, onafhankelijk is van haar voorouders. De veranderlijke is, conditioneel op haar moeder, namelijk ook onafhankelijk van haar zussen en van de nakomelingen van die zussen. Anders gezegd is het model voor een veranderlijke, conditioneel op haar moeder, onafhankelijk van alle veranderlijken die niet de moeder of één van de nakomelingen van deze veranderlijke zijn.

Deze laatste onafhankelijkheidsvoorwaarde is in feite de Markovvoorwaarde die we in paragraaf 2.4 reeds introduceerden. In zijn hier voorgestelde vorm is die niet alleen geldig voor een Markovketen, maar ook voor meer algemene Bayesiaanse netwerken [5], waarvan de Markovketen en de in dit hoofdstuk besproken HMM eenvoudige voorbeelden zijn.

3.3 Opbouw van een globaal model

Aan de hand van de lokale modellen voor de toestandsveranderlijken en de observaties, willen we nu een globaal model opbouwen. We voeren hiervoor opnieuw de conditionele previsies $P_k(\cdot|X_{k-1})$ in, die ditmaal gedefinieerd zijn voor alle $f \in \mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$ en alle $k \in \{1, \dots, n\}$. Voor elke $f \in \mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$ en elke $x_{k-1} \in X_{k-1}$ is $P_k(f|x_{k-1})$ de verwachtingswaarde van f conditioneel op de gebeurtenis $X_{k-1} = x_{k-1}$. Merk op dat we ook hier de conditionele previsie $P_1(\cdot|X_0)$ identificeren met het globale model $P_1(\cdot)$, omdat de veranderlijke X_0 slechts één enkele waarde x_0 kan aannemen en dus zeker is.



We zijn nu op zoek naar recursieve betrekkingen zoals vergelijking (2.13) die ons toelaten om $P_k(\cdot|X_{k-1})$ te bepalen aan de hand van $P_{k+1}(\cdot|X_k)$. In tegenstelling tot bij de Markovketen, hangt nu niet alleen het model $P_{k+1}(\cdot|X_k)$ af van de toestand van de veranderlijke X_k , maar ook het model $S_k(\cdot|X_k)$. We voeren daarom voor alle $k \in \{1, \dots, n\}$ een bijkomende conditionele previsie $E_k(\cdot|X_k)$ in, die de modellen $P_{k+1}(\cdot|X_k)$ en $S_k(\cdot|X_k)$ samenvoegt. $E_k(\cdot|X_k)$ geeft voor alle $f \in \mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n})$ de verwachtingswaarde van f , conditioneel op de waarde van X_k .

We weten uit paragraaf 3.2 dat conditioneel op X_k – dus van zodra de waarde van X_k gekend is –, $X_{k+1:n}$ en $O_{k+1:n}$ onafhankelijk zijn van O_k . Uit de gekende productwet voor verwachtingswaarden van onafhankelijke veranderlijken volgt dan dat voor alle $k \in \{1, \dots, n-1\}$, alle $f \in \mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k+1:n})$ en alle $g \in \mathcal{L}(\mathcal{O}_k)$

$$E_k(fg|X_k) = S_k(g|X_k)P_{k+1}(f|X_k). \quad (3.1)$$

Merk op dat we er hier impliciet van uit gegaan zijn dat fg behoort tot het domein van $E_k(\cdot|X_k)$, namelijk $\mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n})$. Hoewel dit intuïtief duidelijk is, is het toch nuttig om even formeel te vermelden dat dit komt omdat zowel f als g door *cilindrische extensie* kunnen uitgebreid worden zodat ze gedefinieerd zijn op $\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n}$, in plaats van op hun eigen, kleinere domeinen. Hun product is dan logischerwijs ook gedefinieerd op $\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n}$ en behoort dus tot $\mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n})$.

De cilindrische extensie van een functie $f_{\mathcal{X}}$ op \mathcal{X} is de functie $f_{\mathcal{X} \times \mathcal{Y}}$ op $\mathcal{X} \times \mathcal{Y}$, die we voor alle $x \in \mathcal{X}$ en alle $y \in \mathcal{Y}$ definiëren als

$$f_{\mathcal{X} \times \mathcal{Y}}(x, y) = f_{\mathcal{X}}(x).$$

We zullen deze cilindrische extensie in het vervolg van deze masterproef vaak impliciet gebruiken.

Voor $k = n$ kunnen we vergelijking (3.1) niet toepassen. In dat geval zien we dat

$$E_n(\cdot|X_n) = S_n(\cdot|X_n). \quad (3.2)$$

Tot slot zijn we op zoek naar een recursieve betrekking waarmee we de previsie $P_k(\cdot|X_{k-1})$ kunnen bepalen aan de hand van $E_k(\cdot|X_k)$. De relatie tussen deze twee previsions is volkomen analoog aan de relatie tussen $P_k(\cdot|X_{k-1})$ en $P_{k+1}(\cdot|X_k)$ bij een Markovketen. Naar analogie met de in paragraaf 2.6.2 afgeleide uitdrukking vinden we dus onmiddellijk de onderstaande recursieve betrekking, die geldig is voor alle $k \in \{1, \dots, n\}$ en alle $f \in \mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$.

$$P_k(f|X_{k-1}) = Q_k(E_k(f(X_k, X_{k+1:n})|X_k)|X_{k-1}) \quad (3.3)$$

Voor praktische berekeningen is het handig om in te zien dat we dit ook kunnen schrijven als

$$P_k(f|X_{k-1}) = Q_k\left(\sum_{z_k \in \mathcal{X}_k} \mathbb{I}_{\{z_k\}} E_k(f(z_k, X_{k+1:n})|z_k) \Big| X_{k-1}\right).$$

3.4 Het Viterbi-algoritme

3.4.1 Probleemstelling

Nu we een globaal model hebben opgebouwd kunnen we dit model gebruiken om allerhande problemen op te lossen. In het kader van deze masterproef zullen we ons beperken tot één specifiek maar belangrijk probleem: als we de waarde $o_{1:n}$ van alle observaties $O_{1:n}$ kennen, welke waarde $x_{1:n}$ van de toestandssequentie $X_{1:n}$ heeft dan de grootste waarschijnlijkheid om opgetreden te zijn. Of ruwweg gezegd: welk weer was het vermoedelijk de voorbije week als ik weet welke dagen mijn deur geklemd heeft.

Een zeer efficiënte wijze om dit probleem op te lossen is het Viterbi-algoritme. Hoewel dit algoritme in de literatuur meestal aan de hand van waarschijnlijkheden wordt beschreven, kiezen wij er hier voor om te werken met verwachtingswaarden. Dit maakt de omschrijving iets omslachtiger, maar heeft als grote voordeel dat we zo nu al vertrouwd raken met de gebruikte terminologie. Als we later in hoofdstuk 5 overschakelen op imprecieze hidden Markovmodellen, dan zal het noodzakelijk worden om te werken met verwachtingswaarden, en zal de hier geïntroduceerde terminologie de overgang van precieze naar imprecieze modellen veel vlotter doen verlopen.

De eerste stap in het bepalen van de meest waarschijnlijke toestandssequenties is het conditioneren van het globaal model van de HMM op de waargenomen observaties. Als we vervolgens in dit conditioneel model de indicator voor een toestandssequentie $x_{1:n}$ invullen,¹ dan geeft ons dit de waarschijnlijkheid van deze toestandssequentie, conditioneel op de waargenomen observaties $o_{1:n}$. Met de regel van Bayes vinden we hiervoor makkelijk dat

$$P_1(\{x_{1:n}\}|o_{1:n}) = P_1(\mathbb{I}_{\{x_{1:n}\}}|o_{1:n}) = \frac{P_1(\mathbb{I}_{\{x_{1:n}\}}\mathbb{I}_{\{o_{1:n}\}})}{P_1(\{o_{1:n}\})}. \quad (3.4)$$

Merk op dat de bovenstaande uitdrukking pas goed gedefinieerd is als $P_1(\{o_{1:n}\}) \neq 0$. We zullen er in het vervolg van dit hoofdstuk van uitgaan dat dit inderdaad zo is. Dit is een logische veronderstelling, want als $P_1(\{o_{1:n}\})$ gelijk is aan nul, dan wil dat zeggen dat het model de kans dat we de observaties $o_{1:n}$ waarnemen gelijk aan nul acht. We hebben deze observaties echter wel degelijk waargenomen, en dus moeten we ons in dat geval de vraag stellen of we niet beter een ander model beschouwen. We gaan er in wat volgt dus van uit dat $P_1(\{o_{1:n}\}) \neq 0$ en dat de bovenstaande uitdrukking voor de conditionele waarschijnlijkheid van een toestandssequentie dus goed gedefinieerd is.

Aangezien we niet geïnteresseerd zijn in de exacte waarde van vergelijking (3.4), maar enkel willen weten welke toestandssequenties het meest waarschijnlijk zijn, hoeven we ons van de noemer $P_1(\{o_{1:n}\})$ – die voor een gegeven observatie constant is en door het hierboven gezegde positief verondersteld wordt – niets aan te trekken. Het volstaat om een toestandssequentie te zoeken waarvoor de uitdrukking $P_1(\mathbb{I}_{\{x_{1:n}\}}\mathbb{I}_{\{o_{1:n}\}})$ maximaal

1. Voor het gemak van notatie zullen we $\{x_{1:n}\}$ soms identificeren met $\mathbb{I}_{\{x_{1:n}\}}$.

wordt. We noemen zo'n sequentie optimaal en noteren ze als $\hat{x}_{1:n}$. De verzameling van alle optimale sequenties wordt gedefinieerd door

$$\hat{x}_{1:n} \in \text{opt}(\mathcal{X}_{1:n}|o_{1:n}) \Leftrightarrow (\forall x_{1:n} \in \mathcal{X}_{1:n}) P_1(\mathbb{I}_{\{\hat{x}_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}) \geq P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}). \quad (3.5)$$

Al deze optimale sequenties zijn even waarschijnlijk – de waarde van $P_1(\mathbb{I}_{\{\hat{x}_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}})$, en dus ook van $P_1(\{\hat{x}_{1:n}\}|o_{1:n})$, is voor alle $\hat{x}_{1:n} \in \text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ dezelfde – en we noemen ze daarom indifferent.²

Voor praktische situaties is men vaak slechts in één oplossing geïnteresseerd. Men kan dan van zodra men één sequentie gevonden heeft die aan bovenstaande definitie voldoet, stoppen met zoeken, aangezien de andere elementen van $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ niet beter zullen zijn, maar even goed. Zo'n willekeurig gekozen oplossing noteren we als $\text{opt}^*(\mathcal{X}_{1:n}|o_{1:n})$. In tegenstelling tot de verzameling $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ verwijst $\text{opt}^*(\mathcal{X}_{1:n}|o_{1:n})$ dus slechts naar één enkele sequentie.

Stel dat we definitie (3.5) – bijvoorbeeld per ongeluk – zouden toepassen voor een model waarvoor $P_1(\{o_{1:n}\})$ gelijk is aan nul. De gebruikte definitie kan dan niet meer geïnterpreteerd worden als een voorwaarde opdat $P_1(\{\hat{x}_{1:n}\}|o_{1:n})$ maximaal zou zijn. In zo'n geval weten we voor alle $x_{1:n} \in \mathcal{X}_{1:n}$ dat

$$0 \leq P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}) \leq P_1(\mathbb{I}_{\{o_{1:n}\}}) = P_1(\{o_{1:n}\}) = 0,$$

en dus dat $P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}) = 0$ voor alle $x_{1:n} \in \mathcal{X}_{1:n}$, waardoor alle toestandssequenties $x_{1:n} \in \mathcal{X}_{1:n}$ volgens de – nu eigenlijk niet toepasbare – definitie (3.5) optimaal zouden zijn. Als het dus zou voorkomen dat we definitie (3.5) onterecht toepassen, dan kan deze situatie makkelijk opgemerkt en opgevangen worden, en we kunnen er dus in het vervolg van deze uiteenzetting van uit gaan dat definitie (3.5) geldig is.

Het komt er nu op aan om $P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}})$ eenvoudig te kunnen bepalen. Hiervoor is het belangrijk om in te zien dat een indicator van een sequentie gelijk is aan het product van de indicatoren van de individuele elementen van deze sequentie, en dus bijvoorbeeld dat

$$\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}} = \prod_{i=1}^n \mathbb{I}_{\{x_i\}} \prod_{i=1}^n \mathbb{I}_{\{o_i\}}.$$

Met de vergelijkingen (3.3)–(3.2) vinden we nu dat

$$\begin{aligned} P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}) &= Q_1(E_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}} | X_1)) \\ &= Q_1(\mathbb{I}_{\{x_1\}} E_1(\mathbb{I}_{\{x_{2:n}\}} \mathbb{I}_{\{o_{1:n}\}} | x_1)) \\ &= Q_1(\{x_1\}) E_1(\mathbb{I}_{\{x_{2:n}\}} \mathbb{I}_{\{o_{1:n}\}} | x_1) \\ &= Q_1(\{x_1\}) S_1(\{o_1\} | x_1) P_2(\mathbb{I}_{\{x_{2:n}\}} \mathbb{I}_{\{o_{2:n}\}} | x_1). \end{aligned}$$

2. Strikt genomen zijn het niet de sequenties die indifferent zijn, maar zijn wij indifferent (onverschillig) bij een keuze tussen deze sequenties.

We kunnen deze uitdrukking op analoge wijze verder blijven uitwerken en vinden zo tot slot dat

$$P_1(\mathbb{I}_{\{x_{1:n}\}} \mathbb{I}_{\{o_{1:n}\}}) = Q_1(\{x_1\}) \prod_{i=2}^n Q_i(\{x_i\} | x_{i-1}) \prod_{i=1}^n S_i(\{o_i\} | x_i). \quad (3.6)$$

Hoewel dit een eenvoudige uitdrukking is, ligt het op het eerste gezicht niet zo voor de hand om de optimale sequenties $\hat{x}_{1:n}$ te vinden waarvoor ze maximaal wordt. We zouden de uitdrukking (3.6) voor alle mogelijke sequenties $x_{1:n}$ kunnen evalueren en vervolgens rechtstreeks definitie (3.5) toepassen, maar het aantal mogelijke sequenties is exponentieel in de lengte van de sequentie, waardoor de rekentijd zeer lang zou worden. Het is daarom nodig om dit probleem op een verstandigere manier aan te pakken. Het Viterbi-algoritme doet dit zeer elegant door middel van recursie.

3.4.2 Twee verschillende versies van het Viterbi-algoritme

Het Viterbi-algoritme [4, 6] is een efficiënte, recursieve oplossing van het hierboven voorgestelde probleem. Vooraleer we dit algoritme zullen bespreken, is het belangrijk om te vermelden en in te zien dat er twee verschillende versies van dit algoritme bestaan.

In de literatuur wordt meestal een variant beschreven die slechts op zoek is naar één oplossing $\text{opt}^*(\mathcal{X}_{1:n} | o_{1:n})$ in plaats van de volledige verzameling $\text{opt}(\mathcal{X}_{1:n} | o_{1:n})$. In die variant wordt het bestaan van meerdere oplossingen vaak zelfs stilzwijgend genegeerd. Er zijn goede redenen waarom auteurs dit doen. Ten eerste zijn de verschillende oplossingen even goed en moet je er voor praktische situaties uiteindelijk toch gewoon één uit kiezen. Ten tweede zal het in de praktijk maar zelden voorkomen dat er meerdere oplossingen zijn; het moet al zeer “toevallig” zijn dat de waarschijnlijkheid van verschillende sequenties precies gelijk is en dat deze sequenties dan ook nog eens de meest waarschijnlijke zijn. Een derde belangrijke reden is natuurlijk dat het makkelijker is om één oplossing te zoeken, dan om ze allemaal te bepalen.

Ondanks al deze redenen wil ik er in deze masterproef toch de aandacht op vestigen dat er in principe meerdere oplossingen mogelijk zijn en beschrijf ik daarom een variant van het Viterbi-algoritme die in staat is om al die oplossingen te bepalen. Ik heb hier verschillende redenen voor. Een eerste reden is louter principieel. Ook al is men slechts geïnteresseerd in één oplossing, toch vind ik het belangrijk dat men weet dat er eigenlijk meerdere oplossingen kunnen zijn.³ Ten tweede kan het soms wel degelijk nuttig zijn om die verschillende even goede oplossingen allemaal te kunnen bepalen. Men hoeft hier namelijk niet altijd willekeurig een oplossing uit te kiezen. Men kan ook een keuze maken op basis van criteria die niet wiskundig uitgedrukt kunnen worden, zoals bijvoorbeeld persoonlijke voorkeur. Een laatste en belangrijke reden om hier de aandacht te vestigen op het theoretische bestaan van meerdere oplossingen, is omdat op die manier de analogie met het verderop in deze masterproef uiteengezette EstiMMM-algoritme duidelijker wordt.

3. Dit kadert in het algemene principe dat mensen pas gefundeerd een keuze kunnen maken als ze voldoende geïnformeerd zijn.

3.4.3 Een versie die alle oplossingen bepaalt

In deze paragraaf beschrijf ik het Viterbi-algoritme in zijn meest algemene en correcte vorm, namelijk die waarbij de verzameling van alle mogelijke oplossingen bepaald wordt. In een volgende paragraaf zal ik vervolgens nog even kort ingaan op de meer eenvoudige variant die slechts één oplossing zoekt.

In plaats van meteen op zoek te gaan naar de optimale sequenties voor het globale model $P_1(\cdot)$, zullen we voor alle $k \in \{1, \dots, n\}$ en alle $z_{k-1} \in \mathcal{X}_{k-1}$ de verzameling van de optimale sequenties bepalen die hoort bij het model $P_k(\cdot | z_{k-1})$.

$$\begin{aligned} \hat{x}_{k:n} &\in \text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) \\ &\Leftrightarrow (\forall x_{k:n} \in \mathcal{X}_{k:n}) P_k(\mathbb{I}_{\{\hat{x}_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}) \geq P_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}) \end{aligned} \quad (3.7)$$

Door hiervoor te beginnen bij $k = n$ en vervolgens steeds de oplossing voor kleinere k te bepalen op basis van het resultaat voor $k + 1$, bekomen we uiteindelijk recursief de gezochte verzameling $\text{opt}(\mathcal{X}_{1:n} | o_{1:n}) = \text{opt}(\mathcal{X}_{1:n} | z_0, o_{1:n})$ voor het globale model.

Opnieuw zullen we de notatie $\text{opt}^*(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ gebruiken om te verwijzen naar een willekeurig element van de verzameling $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$.

Voor $k = n$ beschouwen we het model $P_n(\cdot | z_{n-1})$. Voor elke $z_{n-1} \in \mathcal{X}_{n-1}$ vinden we analoog aan vergelijking (3.6) dat

$$P_n(\mathbb{I}_{\{x_n\}} \mathbb{I}_{\{o_n\}} | z_{n-1}) = Q_n(\{x_n\} | z_{n-1}) S_n(\{o_n\} | x_n).$$

We kunnen de optimale elementen $\hat{x}_n \in \text{opt}(\mathcal{X}_n | z_{n-1}, o_n)$ nu makkelijk bepalen door simpelweg voor alle $x_n \in \mathcal{X}_n$ de corresponderende waarde te berekenen en vervolgens definitie (3.7) toe te passen voor $k = n$.

Om voor een vaste $z_{k-1} \in \mathcal{X}_{k-1}$ de optimale sequenties $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ te bepalen, maken we gebruik van recursie, en veronderstellen we dus dat we voor alle $z_k \in \mathcal{X}_k$ over de verzameling $\text{opt}(\mathcal{X}_{k+1:n} | z_k, o_{k+1:n})$ beschikken.

Met vergelijkingen (3.3) en (3.1) vinden we meteen dat

$$\begin{aligned} P_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}) \\ = Q_k(\{x_k\} | z_{k-1}) S_k(\{o_k\} | x_k) P_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | x_k), \end{aligned} \quad (3.8)$$

waarmee we de onderstaande stelling kunnen bewijzen.

Stelling 3.1 (Optimaliteitsprincipe)

Kies $k \in \{1, \dots, n-1\}$, $z_{k-1} \in \mathcal{X}_{k-1}$, $\hat{x}_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$. Als $Q_k(\{\hat{x}_k\} | z_{k-1}) > 0$ en $S_k(\{o_k\} | \hat{x}_k) > 0$ dan geldt er dat:

$$\hat{x}_{k:n} \in \text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) \Rightarrow \hat{x}_{k+1:n} \in \text{opt}(\mathcal{X}_{k+1:n} | \hat{x}_k, o_{k+1:n}).$$

Bewijs

Fixeer een $k \in \{1, \dots, n-1\}$, $z_{k-1} \in \mathcal{X}_{k-1}$, $\hat{x}_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$. We nemen aan dat $\hat{x}_{k+1:n}$ geen element is van $\text{opt}(\mathcal{X}_{k+1:n} | \hat{x}_k, o_{k+1:n})$ en tonen aan dat hieruit volgt dat $\hat{x}_{k:n}$ geen element is van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ als $Q_k(\{\hat{x}_k\} | z_{k-1}) > 0$ en $S_k(\{o_k\} | \hat{x}_k) > 0$.

Uit de veronderstelling en definitie (3.7) volgt dat er een $x_{k+1:n} \in \mathcal{X}_{k+1:n}$ is waarvoor $P_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | \hat{x}_k) > P_{k+1}(\mathbb{I}_{\{\hat{x}_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | \hat{x}_k)$. We verlengen deze sequentie $x_{k+1:n}$ door er vooraan de toestand \hat{x}_k aan toe te voegen, en bekommen zo de sequentie $x_{k:n}$, wat impliceert dat $\hat{x}_k = x_k$. Met behulp van vergelijking (3.8) vinden we dan dat

$$\begin{aligned} P_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}) &= Q_k(\{x_k\} | z_{k-1}) S_k(\{o_k\} | x_k) P_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | x_k) \\ &= Q_k(\{\hat{x}_k\} | z_{k-1}) S_k(\{o_k\} | \hat{x}_k) P_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | \hat{x}_k) \\ &> Q_k(\{\hat{x}_k\} | z_{k-1}) S_k(\{o_k\} | \hat{x}_k) P_{k+1}(\mathbb{I}_{\{\hat{x}_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | \hat{x}_k) \\ &= P_k(\mathbb{I}_{\{\hat{x}_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}), \end{aligned}$$

waarbij de ongelijkheid geldt omdat $Q_k(\{\hat{x}_k\} | z_{k-1}) > 0$ en $S_k(\{o_k\} | \hat{x}_k) > 0$. Hiermee is bewezen dat $\hat{x}_{k:n}$ inderdaad geen element is van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$. \square

Een direct gevolg van dit optimaliteitsprincipe is dat

$$\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) \subseteq \text{mog}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}), \quad (3.9)$$

waarbij $\text{mog}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ de verzameling is van alle mogelijke optimale sequenties, die door stelling 3.1 nog steeds een element kunnen zijn van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$.

Om een uitdrukking voor deze verzameling op te stellen hebben we de operator \oplus nodig. Deze operator concateneert twee sequenties ($z_k \oplus z_{k+1:n}$ is dus hetzelfde als $z_{k:n}$). Als één van deze beide sequenties een verzameling van sequenties is, dan vinden we na concatenatie op voor de hand liggende wijze opnieuw een verzameling, ditmaal van geconcateneerde sequenties. Met deze operator definiëren we $\text{mog}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ nu als volgt:

$$\begin{aligned} &\text{mog}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) \\ &= \left(\bigcup_{z_k \in \text{Pos}_k(z_{k-1})} z_k \oplus \text{opt}(\mathcal{X}_{k+1:n} | z_k, o_{k+1:n}) \right) \cup \left(\bigcup_{z_k \notin \text{Pos}_k(z_{k-1})} z_k \oplus \mathcal{X}_{k+1:n} \right). \quad (3.10) \end{aligned}$$

De verzameling $\text{Pos}_k(z_{k-1}) \in \mathcal{X}_k$ wordt hierbij gegeven door

$$z_k \in \text{Pos}_k(z_{k-1}) \Leftrightarrow Q_k(\{z_k\} | z_{k-1}) > 0 \text{ en } S_k(\{o_k\} | z_k) > 0. \quad (3.11)$$

Bovendien weten we voor alle $z_k \notin \text{Pos}_k(z_{k-1})$ door vergelijking (3.8) dat de waarde van $P_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1})$ voor alle $x_{k:n} \in z_k \oplus \mathcal{X}_{k+1:n}$ gelijk is aan nul. Als een dergelijke sequentie $x_{k:n}$ optimaal zou zijn, dan volgt uit definitie (3.7) voor alle $z_{k:n} \in \mathcal{X}_{k:n}$ dat

$P_k(\mathbb{I}_{\{z_{k:n}\}}\mathbb{I}_{\{o_{k:n}\}}|z_{k-1}) = 0$. In dat geval geldt dus dat $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) = \mathcal{X}_{k:n}$. Er geldt dan echter ook dat

$$P_1(\{o_{k:n}\}) = P_1\left(\sum_{z_{k:n} \in \mathcal{X}_{k:n}} \mathbb{I}_{\{z_{k:n}\}}\mathbb{I}_{\{o_{k:n}\}}\right) = \sum_{z_{k:n} \in \mathcal{X}_{k:n}} P_1(\mathbb{I}_{\{z_{k:n}\}}\mathbb{I}_{\{o_{k:n}\}}) = 0.$$

We beargumenteerden in paragraaf 3.4.1 al dat we dit geval uitsluiten en dus kunnen we er voor het vervolg van deze uiteenzetting van uitgaan dat voor alle $z_k \notin \text{Pos}_k(z_{k-1})$ de verzameling $z_k \oplus \mathcal{X}_{k+1:n}$ niet tot $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ behoort. Hieruit volgt samen met vergelijking (3.9) en definitie (3.10) dat

$$\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \subseteq \bigcup_{z_k \in \text{Pos}_k(z_{k-1})} z_k \oplus \text{opt}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n}). \quad (3.12)$$

We weten verder voor een vaste $z_k \in \text{Pos}_k(z_{k-1})$ ook dat voor alle $\hat{x}_{k+1:n}$ die tot de verzameling $\text{opt}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$ behoren, de waarde van $P_{k+1}(\mathbb{I}_{\{\hat{x}_{k+1:n}\}}\mathbb{I}_{\{o_{k+1:n}\}}|z_k)$ door definitie (3.7) identiek is. Hieruit kunnen we via vergelijking (3.8) en definitie (3.7) voor alle $z_k \in \text{Pos}_k(z_{k-1})$ afleiden dat $z_k \oplus \text{opt}^*(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$ behoort tot $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ als en slechts als de hele verzameling $z_k \oplus \text{opt}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$ tot $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ behoort.

Uit het bovenstaande en vergelijking (3.12) kunnen we besluiten dat de verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ makkelijk kan bepaald worden. Definitie (3.7) is namelijk equivalent met

$$\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) = \bigcup_{\hat{z}_k \in Z_k^*(z_{k-1})} \hat{z}_k \oplus \text{opt}(\mathcal{X}_{k+1:n}|\hat{z}_k, o_{k+1:n}). \quad (3.13)$$

De verzameling $Z_k^*(z_{k-1})$ wordt hierbij gedefinieerd als die $\hat{z}_k \in \text{Pos}_k(z_{k-1})$ waarvoor voor alle $z_k \in \text{Pos}_k(z_{k-1})$ geldt dat

$$P_k\left(\mathbb{I}_{\{z_k^{\oplus*}\}}\mathbb{I}_{\{o_{k:n}\}}|z_{k-1}\right) \geq P_k\left(\mathbb{I}_{\{\hat{z}_k^{\oplus*}\}}\mathbb{I}_{\{o_{k:n}\}}|z_{k-1}\right),$$

waarbij we voor alle $z_k \in \mathcal{X}_k$ de sequentie $z_k^{\oplus*} = z_k \oplus \text{opt}^*(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$ beschouwd hebben en de uitdrukkingen van de vorm $P_k(\mathbb{I}_{\{x_{k:n}\}}\mathbb{I}_{\{o_{k:n}\}}|z_{k-1})$ recursief berekenen met vergelijking (3.8).

3.4.4 Een versie die slechts één oplossing bepaalt

De eenvoudiger variant van het Viterbi-algoritme, die slechts één oplossing bepaalt, kan – in zeer sumiere vorm – als volgt omschreven worden.

Voor $k = n$ gebruiken we opnieuw rechtstreeks definitie (3.7) om een optimaal element te vinden, maar ditmaal zijn we voor elke $z_{n-1} \in \mathcal{X}_{n-1}$ slechts in één willekeurig optimaal element $\text{opt}^*(\mathcal{X}_n|z_{n-1}, o_n)$ geïnteresseerd.

Voor alle $k \in \{1, \dots, n-1\}$ maken we gebruik van recursie en veronderstellen we dat we voor elke $z_k \in \mathcal{X}_k$ over een optimale sequentie $\text{opt}^*(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$ beschikken. Hiermee kunnen we als volgt voor elke $z_{k-1} \in \mathcal{X}_{k-1}$ een optimale sequentie $\text{opt}^*(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ bepalen:

$$\text{opt}^*(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) = \hat{z}_k^* \oplus \text{opt}^*(\mathcal{X}_{k+1:n}|\hat{z}_k^*, o_{k+1:n}),$$

waarbij \hat{z}_k^* een willekeurig gekozen element van $Z_k^*(z_{k-1})$ is.

Door deze recursiebetrekking steeds opnieuw toe te passen, bekomen we uiteindelijk voor $k = 1$ de gezochte optimale sequentie $\text{opt}^*(\mathcal{X}_{1:n}|o_{1:n}) = \text{opt}^*(\mathcal{X}_{1:n}|z_0, o_{1:n})$.

Voor een uitgebreidere uiteenzetting van deze variant van het algoritme, die zich niet baseert op de algemenere versie, maar rechtstreeks het eenvoudige Viterbi-algoritme beschrijft, verwijzen we naar [4] en [6].

3.4.5 Bespreking van de complexiteit

Voor $k = n$ is voor elke $z_{n-1} \in \mathcal{X}_{n-1}$ het aantal bewerkingen dat nodig is om de verzameling $\text{opt}(\mathcal{X}_n|z_{n-1}, o_n)$ te bepalen lineair in het aantal mogelijke toestanden. We kunnen bijvoorbeeld voor alle $x_n \in \mathcal{X}_n$ de waarde van $P_n(\mathbb{I}_{\{x_n\}}\mathbb{I}_{\{o_n\}}|z_{n-1})$ bepalen, vervolgens hiervan een maximum bepalen en tot slot de lijst nog eens overlopen om te kijken of er niet meerdere elementen zijn waarvoor dit maximum bereikt wordt. Deze drie stappen zijn alle lineair in het aantal toestanden, en dus is hun opeenvolging dit ook. Het bepalen van al deze verzamelingen $\text{opt}(\mathcal{X}_n|z_{n-1}, o_n)$, voor elke $z_{n-1} \in \mathcal{X}_{n-1}$, is dan natuurlijk kwadratisch in het aantal toestanden.

Voor de overige k is de stap met de grootste, en dus meest bepalende, complexiteit het opbouwen van de verzamelingen $Z_k^*(z_{k-1})$. De complexiteit van het bepalen van één zulke verzameling is opnieuw lineair in het aantal mogelijke toestanden. Dit komt doordat we eerst de verzameling $\text{Pos}_k(z_{k-1})$ moeten bepalen, wat lineair is in het aantal toestanden, en we vervolgens in essentie opnieuw de maxima moeten zoeken van een lijst waarvan het aantal elementen gelijk is aan het aantal toestanden, en waarvan de waarde van de elementen in een constant aantal bewerkingen bepaald kan worden. De reden waarom we de waarde van zo'n element $P_k(\mathbb{I}_{\{x_{k:n}\}}\mathbb{I}_{\{o_{k:n}\}}|z_{k-1})$ in een constant aantal bewerkingen kunnen bepalen, is dat we hiervoor de recursieve betrekking (3.8) kunnen gebruiken en dat we de waarde van $P_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}}\mathbb{I}_{\{o_{k+1:n}\}}|x_k)$ al berekend hebben in de vorige stap. Aangezien het opbouwen van één verzameling $Z_k^*(z_{k-1})$ lineair is in het aantal toestanden, is het opbouwen van al deze verzamelingen, voor elke $z_{k-1} \in \mathcal{X}_{k-1}$, kwadratisch in het aantal toestanden.

Om het hele algoritme uit te voeren, moeten we dus voor alle $k \in \{1, \dots, n\}$ een aantal bewerkingen uitvoeren dat kwadratisch is in het aantal mogelijk toestanden. Bijgevolg is de complexiteit van het algoritme lineair in de lengte van de beschouwde sequenties en kwadratisch in het aantal toestanden.

Op het eerste gezicht lijkt er geen verschil in complexiteit te zijn tussen de eenvoudige variant van het Viterbi-algoritme, die slechts één oplossing zoekt, en de algemenere

variant, die de hele verzameling van oplossingen bepaalt. Dit is echter niet correct. Het uitschrijven van alle oplossingen (en dus ook het algoritme, aangezien het uitschrijven van de oplossingen daar deel van uit maakt) is namelijk lineair in het aantal bekomen oplossingen. Stel u bijvoorbeeld het uitzonderingsgeval voor waarbij alle mogelijke sequenties even waarschijnlijk zijn. In dit geval wordt de lineariteit in het aantal oplossingen duidelijk de meest bepalende factor. In de variant waarbij men slechts één oplossing zoekt is dit, om voor de hand liggende redenen, niet het geval.

3.4.6 Implementatie in Python

Ten behoeve van de toepassing in hoofdstuk 8, werd de in paragraaf 3.4.3 besproken versie van het Viterbi-algoritme geïmplementeerd in de programmeertaal Python. Deze code bevat tevens een optie om de eenvoudiger versie van paragraaf 3.4.4 te gebruiken. Een neerslag van de code is terug te vinden in bijlage C.

Hoofdstuk 4

Coherente onderprevisies

“Free yourself from the rigid conduct of tradition and open yourself to the new forms of probability.”

– Hans Bender

4.1 Inleiding

Stel, u bent een man. Als er in uw familie in het verleden een geval van teelbalkanker is opgetreden – uw grootvader is er bijvoorbeeld voor behandeld –, dan bent u wellicht geneigd preventief een dokter op te zoeken. Hij stelt u een aantal vragen over uw leefgewoonten – “Bent u al dan niet van plan in de nabije toekomst de ronde van Frankrijk te proberen winnen?” –, informeert zich over de eventuele gevallen van teelbalkanker die in uw familie zijn opgetreden, en zegt u om volgende week nog eens terug te komen. In de tussentijd zal hij voor u uitzoeken of het al dan niet een goed idee is om verdere, dure en mogelijk risicovolle onderzoeken te laten uitvoeren.

Nadat u vertrokken bent, probeert de dokter in te schatten welk risico u loopt op teelbalkanker. Aangezien uw grootvader teelbalkanker had, geeft dat u volgens zijn ervaringen met andere patiënten 10% kans om teelbalkanker te krijgen. U bent echter ook van plan om in de nabije toekomst de ronde van Frankrijk te proberen winnen, wat u volgens hem 20% kans geeft op teelbalkanker. Over het risico dat u loopt als u enorme wielierambities heeft én u een grootvader heeft die voor teelbalkanker is behandeld durft hij echter geen uitspraken te doen. Bovendien moet de dokter ook nog rekening houden met de gegevens die u hem heeft gegeven over uw omgang met vrouwen, andere leefgewoonten, ...

Het wordt de dokter iets te ingewikkeld om met al deze gegevens op correcte wijze rekening te houden en hij besluit daarom om er zijn vriend de wiskundige bij te halen. Hij geeft zijn vriend alle gegevens, waaronder ook de door hemzelf ingeschatte kansen, waarna de wiskundige er de waarschijnlijkheidsrekening op los laat. Na enig rekenwerk besluit hij dat het risico dat u loopt op teelbalkanker zodanig klein is dat het geen goed idee is om dure, mogelijk risicovolle onderzoeken te laten uitvoeren.

Mijn kennis over geneeskunde is beperkt. De uwe misschien ook. Toch vermoed ik dat de meeste mensen, ongeacht hun kennis over geneeskunde, het over het volgende eens zullen zijn. Wat als de dokter niet zeker is over uw kans op teelbalkanker als u van plan bent de ronde van Frankrijk te gaan winnen? Wat als hij eerlijk moet toegeven dat het net zo goed 21% zou kunnen zijn, in plaats van 20%? Wat als de wiskundige uitgaande van die 21% vervolgens zou concluderen dat u beter wel enkele onderzoeken laat uitvoeren? Zou u dan niet willen dat uw dokter u zegt dat hij eigenlijk niet goed weet of u het best extra onderzoeken laat uitvoeren? Ik alvast wel.

In dit hoofdstuk zullen we de theorie van coherente onderprevisies inleiden. Deze theorie laat toe om rekening te houden met een gebrek aan zekerheid over waarschijnlijkheden – “Is het nu 20% of 21%?” – en stelt ons in staat om hier op een coherente manier mee te redeneren. Op die manier kunnen we correcte gevolgtrekking maken uit onvolledige of imprecieze informatie. Uw dokter zal u dan niet altijd adviseren om al dan niet een behandeling te ondergaan, maar soms ook zeggen dat hij het met de voor hem beschikbare informatie eigenlijk niet goed weet. Hoewel toegeven dat men iets niet weet op het eerste zicht minder aantrekkelijk lijkt, ziet u door bovenstaand voorbeeld ongetwijfeld in dat dit soms zeer nuttig kan zijn.

4.2 Imprecieze waarschijnlijkheden

Tot nu toe zijn we er in deze masterproef steeds van uitgegaan dat de waarschijnlijkheden waar we mee rekenden precies gekend waren. De in hoofdstuk 2 ingevoerde massafuncties waren bijvoorbeeld steeds precies. Zoals we in de inleiding tot dit hoofdstuk al aangaven, is dit vaak niet te verantwoorden. Het is in vele gevallen veiliger om toe te geven dat onze initiële kennis beperkt is en vervolgens te beredeneren welke conclusies uit deze imprecieze kennis kunnen getrokken worden. Men spreekt in dit opzicht vaak over robuust redeneren. De op die manier getrokken conclusies zijn veiliger en worden bijvoorbeeld niet beïnvloed door kleine wijzigingen in precieze uitgangsveronderstellingen, die vaak niet van te voren uit te sluiten zijn.

We zullen daarom een imprecies waarschijnlijkheidsmodel opbouwen waarbij we in plaats van één massafunctie, een volledige verzameling \mathcal{M} van massafuncties p gedefinieerd op \mathcal{X} beschouwen [2]. \mathcal{M} is dus een deelverzameling van $\Sigma_{\mathcal{X}}$. De conclusies die we vervolgens zullen trekken, zijn die welke volgen uit elk van de massafuncties in de verzameling \mathcal{M} . Volgen uit verschillende massafuncties, verschillende conclusies, dan blijven we daartussen verder onbeslist.

De verzamelingen van massafuncties die we hiervoor beschouwen zullen steeds convex en gesloten zijn. Een dergelijke convexe gesloten deelverzameling van $\Sigma_{\mathcal{X}}$, noemen we een *credale verzameling*. Eén van de redenen waarom we deze twee voorwaarden opleggen, is omdat dit wiskundig handig is. Er zijn ook andere, beslistheoretische redenen [2, §3.7–3.9], maar een echte diepe verantwoording hiervan zou ons te ver

leiden in het kader van deze masterproef. We zullen ons daarom beperken tot een intuïtieve verklaring.

Als u denkt dat de waarschijnlijkheid van een bepaalde gebeurtenis 20% zou kunnen zijn, maar vindt dat deze ook 30% kan zijn, dan zult u het met me eens zijn dat het niet vreemd is om in dat geval te eisen dat u ook vindt dat het elk percentage tussen 20 en 30 kan zijn. Deze elementaire logica zit vervat in de eis dat de verzameling convex moet zijn.

De voorwaarde dat de verzameling gesloten moet zijn kunnen we als volgt begrijpen. Als u vindt dat een bepaalde kans elk getal *tussen* 20% en 30% kan zijn, dan volgt uit de eis van geslotenheid, dat u ook moet vinden dat de kans 20% of 30% kan zijn. Hoewel deze eis minder makkelijk te verantwoorden is, zal u het met me eens zijn dat het verschil in de praktijk tussen het al dan niet gesloten zijn van dat interval, klein is. Het heeft echter als grote voordeel dat het minimum goed gedefinieerd is. In dit kleine voorbeeld is het minimum van de beschouwde kans bijvoorbeeld 20%. Als we enkel zeggen dat de kans tussen 20% en 30% ligt, zonder dat 20% er zelf bij zit, dan is het begrip minimum niet gedefinieerd. Het opleggen van deze geslotenheid stelt ons bijvoorbeeld verderop in definitie (4.2) in staat om het minimum te kunnen nemen.

4.3 Onderprevisies

Met elke massafunctie $p \in \mathcal{M}$ kunnen we een lineaire previsie (zoals ingevoerd in paragraaf 2.6.1) associëren, gedefinieerd op $\mathcal{L}(\mathcal{X})$ en gegeven door

$$P_p(f) = \sum_{z \in \mathcal{X}} f(z)p(z). \quad (4.1)$$

Deze previsie is de verwachtingswaarde van f die overeenstemt met de massafunctie p . Zoals we in paragraaf 2.6 aantoonden, is een previsie volkomen equivalent met haar corresponderende massafunctie, en kunnen we dus zonder bezwaar redeneren met previsies, in plaats van met waarschijnlijkheden.

Om robuust te redeneren zouden we met elk van de verschillende previsies P_p moeten werken en vervolgens kijken welke gemeenschappelijke besluiten hieruit volgen. Het probleem hiermee is dat het aantal verschillende previsies dat we moeten beschouwen oneindig is, en dat dit dus praktisch niet haalbaar is. We hebben daarom efficiëntere methoden nodig om na te gaan welke gemeenschappelijke besluiten men kan trekken uit verzamelingen van previsies. Er zijn verschillende methoden om dit te doen, waarvan we er hier één zullen bespreken, die we in het vervolg van deze masterproef intensief zullen gebruiken. We voeren een soort overkoepelend model in, waarin alle informatie vervat zit die volgt uit de hele beschouwde verzameling van previsies:

$$\underline{P}_{\mathcal{M}}(f) = \min \{P_p(f) : p \in \mathcal{M}\}. \quad (4.2)$$

We noemen dit model een *onderprevisie* [2]. Uit de definitie volgt dat het model $\underline{P}_{\mathcal{M}}$ voor elke $f \in \mathcal{L}(\mathcal{X})$ de minimale waarde geeft van de verwachtingswaarde $P_p(f)$ die men kan bekomen voor een $p \in \mathcal{M}$.

Op volkomen analoge wijze kunnen we ook een bovenprevisie [2] invoeren.

$$\bar{P}_{\mathcal{M}}(f) = \max \{P_p(f) : p \in \mathcal{M}\} \quad (4.3)$$

We zullen het in de rest van deze masterproef echter vooral over onderprevisies hebben, aangezien beide modellen equivalent zijn. De bovenprevisie die overeenstemt met een credale verzameling \mathcal{M} is namelijk de toegevoegde van de onderprevisie die bij diezelfde \mathcal{M} hoort.

$$\begin{aligned} \bar{P}_{\mathcal{M}}(f) &= \max \{P_p(f) : p \in \mathcal{M}\} \\ &= -\min \{-P_p(f) : p \in \mathcal{M}\} \\ &= -\min \{P_p(-f) : p \in \mathcal{M}\} \\ &= -\underline{P}_{\mathcal{M}}(-f) \end{aligned} \quad (4.4)$$

4.4 Coherentie

Als een op $\mathcal{L}(\mathcal{X})$ gedefinieerde functie \underline{P} voor alle f en g in $\mathcal{L}(\mathcal{X})$ en voor alle $\lambda \geq 0$ voldoet aan de volgende drie axioma's, dan noemen we dit een *coherente* onderprevisie [2].

- (C1) $\underline{P}(f) \geq \min f$
- (C2) $\underline{P}(\lambda f) = \lambda \underline{P}(f)$ [niet-negatieve homogeniteit]
- (C3) $\underline{P}(f + g) \geq \underline{P}(f) + \underline{P}(g)$ [superadditiviteit]

Stelling 4.1

Als we aan de hand van een verzameling \mathcal{M} van massafuncties de onderprevisie $\underline{P}_{\mathcal{M}}$ invoeren via vergelijking (4.2), dan is deze onderprevisie coherente.

Bewijs

Bewijs van axioma C1:

Voor elke $p \in \mathcal{M}$ en alle $f \in \mathcal{L}(\mathcal{X})$ weten we dat

$$P_p(f) = \sum_{z \in \mathcal{X}} f(z)p(z) \geq \sum_{z \in \mathcal{X}} (\min f)p(z) = \min f \sum_{z \in \mathcal{X}} p(z) = \min f,$$

waaruit volgt dat voor alle $f \in \mathcal{L}(\mathcal{X})$

$$\underline{P}_{\mathcal{M}}(f) = \min \{P_p(f) : p \in \mathcal{M}\} \geq \min \{\min f : p \in \mathcal{M}\} = \min f.$$

Bewijs van axioma C2:

Voor elke $p \in \mathcal{M}$, alle $f \in \mathcal{L}(\mathcal{X})$ en alle $\lambda \in \mathbb{R}$ weten we door de homogeniteit van $P_p(\cdot)$ dat

$$P_p(\lambda f) = \lambda P_p(f).$$

Hieruit volgt dat voor alle $f \in \mathcal{L}(\mathcal{X})$ en alle $\lambda \geq 0$

$$\begin{aligned} \underline{P}_{\mathcal{M}}(f) &= \min \{P_p(\lambda f) : p \in \mathcal{M}\} \\ &= \min \{\lambda P_p(f) : p \in \mathcal{M}\} \\ &= \lambda \min \{P_p(f) : p \in \mathcal{M}\} \\ &= \lambda \underline{P}_{\mathcal{M}}(f). \end{aligned}$$

Bewijs van axioma C3:

Voor elke $p \in \mathcal{M}$ en alle f en g in $\mathcal{L}(\mathcal{X})$ weten we door de lineariteit van $P_p(\cdot)$ dat

$$P_p(f + g) = P_p(f) + P_p(g).$$

Hieruit volgt voor alle f en g in $\mathcal{L}(\mathcal{X})$ dat

$$\begin{aligned} \underline{P}_{\mathcal{M}}(f + g) &= \min \{P_p(f + g) : p \in \mathcal{M}\} \\ &= \min \{P_p(f) + P_p(g) : p \in \mathcal{M}\} \\ &\geq \min \{P_p(f) : p \in \mathcal{M}\} + \min \{P_p(g) : p \in \mathcal{M}\} \\ &= \underline{P}_{\mathcal{M}}(f) + \underline{P}_{\mathcal{M}}(g). \end{aligned}$$

□

Met elke op $\mathcal{L}(\mathcal{X})$ gedefinieerde onderprevisie \underline{P} stemt een bovenprevisie \overline{P} overeen die per definitie de toegevoegde is van \underline{P} : voor elke $f \in \mathcal{L}(\mathcal{X})$ is

$$\overline{P}(f) = -\underline{P}(-f). \quad (4.5)$$

We noemen \overline{P} coherent als haar overeenstemmende onderprevisie dit is.

Op basis van de drie coherentie-axioma's kunnen nu heel wat nuttige eigenschappen afgeleid worden. We vermelden diegene waarvan we in het vervolg van deze masterproef gebruik maken. De onderstaande eigenschappen worden bewezen in Bijlage A.

Eigenschap 4.1

Als P een coherente onderprevisie is, gedefinieerd op $\mathcal{L}(\mathcal{X})$, dan volgt hieruit voor alle $f \in \mathcal{L}(\mathcal{X})$ dat $\min f \leq \underline{P}(f) \leq \overline{P}(f) \leq \max f$ en voor alle $\mu \in \mathbb{R}$ dat $\underline{P}(f) = \overline{P}(\mu) = \mu$.

Eigenschap 4.2

Als voor een $f, g \in \mathcal{L}(\mathcal{X})$ geldt dat $f(x) > g(x)$ voor alle $x \in \mathcal{X}$, dan is $\underline{P}(f) > \underline{P}(g)$ voor elke op $\mathcal{L}(\mathcal{X})$ gedefinieerde coherente onderprevisie \underline{P} .

Eigenschap 4.3

Als voor een $f, g \in \mathcal{L}(\mathcal{X})$ geldt dat $f(x) \geq g(x)$ voor alle $x \in \mathcal{X}$, dan is $\underline{P}(f) \geq \underline{P}(g)$ voor elke op $\mathcal{L}(\mathcal{X})$ gedefinieerde coherente onderprevisie \underline{P} .

Eigenschap 4.4

Als \underline{P} een coherente onderprevisie is, gedefinieerd op $\mathcal{L}(\mathcal{X})$, dan volgt hieruit voor elk natuurlijk getal $n \geq 1$, voor willekeurige $f_0, \dots, f_n \in \mathcal{L}(\mathcal{X})$ en $\lambda_0, \dots, \lambda_n \geq 0$ dat $\max(\sum_{j=1}^n \lambda_j G(f_j) - \lambda_0 G(f_0)) \geq 0$. Hierbij definiëren we $G(f) = f - \underline{P}(f)$ voor alle $f \in \mathcal{L}(\mathcal{X})$.

4.5 Terug naar credale verzamelingen

Met elke onderprevisie \underline{P} kunnen we op unieke wijze een verzameling $\mathcal{M}(\underline{P})$ van massafuncties associëren, die we de verzameling van dominerende massafuncties noemen [2]:

$$p \in \mathcal{M}(\underline{P}) \Leftrightarrow (p \in \Sigma_{\mathcal{X}} \text{ en } \forall f \in \mathcal{L}(\mathcal{X}) : P_p(f) \geq \underline{P}(f)). \quad (4.6)$$

Stelling 4.2

De met een onderprevisie \underline{P} geassocieerde verzameling $\mathcal{M}(\underline{P})$ is credaal.

Bewijs

De geslotenheid van de verzameling is triviaal en volgt uit het gebruik van de niet-strikte ongelijkheid \geq in de definitie van $\mathcal{M}(\underline{P})$. Ook de convexiteit is makkelijk in te zien. Beschouw een $p_1, p_2 \in \mathcal{M}(\underline{P})$ en de convexe combinatie $p = \lambda p_1 + (1 - \lambda)p_2$, met $0 \leq \lambda \leq 1$. We vinden dan voor elke $f \in \mathcal{L}(\mathcal{X})$ dat

$$\begin{aligned} P_p(f) &= \sum_{z \in \mathcal{X}} f(z)p(z) = \sum_{z \in \mathcal{X}} f(z)(\lambda p_1(z) + (1 - \lambda)p_2(z)) \\ &= \lambda \sum_{z \in \mathcal{X}} f(z)p_1(z) + (1 - \lambda) \sum_{z \in \mathcal{X}} f(z)p_2(z) = \lambda P_{p_1}(f) + (1 - \lambda)P_{p_2}(f) \\ &\geq \lambda \underline{P}(f) + (1 - \lambda)\underline{P}(f) = \underline{P}(f), \end{aligned}$$

waaruit volgt dat $p \in \mathcal{M}(\underline{P})$, en dus dat $\mathcal{M}(\underline{P})$ een convexe verzameling is. Aangezien $\mathcal{M}(\underline{P})$ een verzameling van massafuncties is die zowel convex als gesloten is, is ze per definitie een credale verzameling. \square

Stelling 4.3

Als de onderprevisie \underline{P} coherent is, dan is $\mathcal{M}(\underline{P})$ een niet-lege verzameling. Bovendien weten we dan dat $\underline{P}_{\mathcal{M}(\underline{P})} = \underline{P}$.

Bewijs

Beschouw voor een vaste, willekeurige $f_0 \in \mathcal{L}(\mathcal{X})$ de verzameling

$$\mathcal{V}(f_0) = \left\{ g \in \mathcal{L}(\mathcal{X}) : g \geq \sum_{j=1}^n \lambda_j G(f_j) - \lambda_0 G(f_0) \text{ met } n \geq 1, f_j \in \mathcal{L}(\mathcal{X}), \lambda_j \geq 0 \right\},$$

waarbij we voor alle $f \in \mathcal{L}(\mathcal{X})$ stellen dat $G(f) = f - \underline{P}(f)$. Men ziet makkelijk in dat dit een convexe verzameling is en dat 0 een element is van $\mathcal{V}(f_0)$. Aangezien alle positieve functies eveneens tot deze verzameling behoren weten we ook dat het inwendige van $\mathcal{V}(f_0)$ niet leeg is.

Stel dat 0 tot het inwendige van $\mathcal{V}(f_0)$ behoort. Dit zou willen zeggen dat er een $\mu > 0$ bestaat waarvoor $-\mu \in \mathcal{V}(f_0)$. Uit eigenschap 4.4 volgt echter voor elke $g \in \mathcal{V}(f_0)$ dat $\max g \geq 0$, en dus zou $-\mu = \max(-\mu) \geq 0$ of dus $\mu \leq 0$. Dit is een contradictie en dus behoort 0 niet tot het inwendige van $\mathcal{V}(f_0)$.

We maken nu gebruik van de ondersteunende hypervlak-stelling voor eindige ruimten [7, Theorem 11.6]. Uit deze stelling volgt onder andere dat als we een convexe deelverzameling C van $\mathcal{L}(\mathcal{X})$ hebben met een niet-leeg inwendige, dat er dan voor elk punt x dat tot C , maar niet tot het inwendige van C behoort, een ondersteunend hypervlak bestaat dat C ondersteunt in x . Dit wil zeggen dat er een $p \in \Sigma(\mathcal{X})$ bestaat die niet identisch nul is en waarvoor voor alle $y \in \mathcal{L}(\mathcal{X})$ geldt dat $\sum_{z \in \mathcal{X}} p(z)y(z) \geq \sum_{z \in \mathcal{X}} p(z)x(z)$.

Stellen we $C = \mathcal{V}(f_0)$, $x = 0$ en $y = g$, dan volgt uit deze stelling dat er een $p \in \Sigma(\mathcal{X})$ bestaat waarvoor voor alle $g \in \mathcal{V}(f_0)$ geldt dat $\sum_{z \in \mathcal{X}} p(z)g(z) \geq 0$. Bovendien weten we voor alle $x \in \mathcal{X}$ dat $\mathbb{I}_{\{x\}}$ tot $\mathcal{V}(f_0)$ behoort, en dus dat $p(x) \geq 0$. Aangezien p niet identisch nul is volgt hieruit dat we p kunnen herschalen tot een massafunctie die behoort tot $\Sigma_{\mathcal{X}}$. Een dergelijke herschaling verandert niets aan de eigenschap dat $\sum_{z \in \mathcal{X}} p(z)g(z) \geq 0$ en aangezien p nu een massafunctie is, is deze eigenschap equivalent met zeggen dat er een $p \in \Sigma_{\mathcal{X}}$ bestaat waarvoor $P_p(g) \geq 0$ voor alle $g \in \mathcal{V}(f_0)$.

Voor alle $f \in \mathcal{L}(\mathcal{X})$ behoort $G(f)$ tot $\mathcal{V}(f_0)$, en dus weten we dat er een $p \in \Sigma_{\mathcal{X}}$ bestaat waarvoor geldt dat $0 \leq P_p(G(f)) = P_p(f - \underline{P}(f)) = P_p(f) - P_p(\underline{P}(f)) = P_p(f) - \underline{P}(f)$, of dus dat $P_p(f) \geq \underline{P}(f)$ voor alle $f \in \mathcal{L}(\mathcal{X})$. Als we dit resultaat combineren met definitie (4.6) dan volgt hieruit dat $\mathcal{M}(\underline{P})$ een niet-lege verzameling is.

Aangezien $-G(f_0) \in \mathcal{V}(f_0)$ weten we dat $0 \leq P_p(-G(f_0)) = -P_p(f_0) + \underline{P}(f_0)$, of dus dat $P_p(f_0) \leq \underline{P}(f_0)$. Maar voor alle $f \in \mathcal{L}(\mathcal{X})$, en dus ook voor f_0 geldt eveneens dat $P_p(f) \geq \underline{P}(f)$. Hieruit besluiten we dat $P_p(f_0) = \underline{P}(f_0)$. Dit is geldig voor elke $f_0 \in \mathcal{L}(\mathcal{X})$ en dus bestaat er voor elke $f_0 \in \mathcal{L}(\mathcal{X})$ een $p \in \mathcal{M}(\underline{P})$, waarvoor geldt dat $P_p(f_0) = \underline{P}(f_0)$. Hieruit volgt dat $\underline{P}_{\mathcal{M}(\underline{P})}(f_0) = \min \{P_p(f_0) : p \in \mathcal{M}(\underline{P})\} \leq \underline{P}(f_0)$ voor alle $f_0 \in \mathcal{L}(\mathcal{X})$. Uit definitie (4.6) en (4.2) kunnen we echter ook afleiden dat $\underline{P}_{\mathcal{M}(\underline{P})}(f_0) \geq \underline{P}(f_0)$, waaruit we besluiten dat $\underline{P}_{\mathcal{M}(\underline{P})}(f_0) = \underline{P}(f_0)$ voor alle $f_0 \in \mathcal{L}(\mathcal{X})$ en dus dat $\underline{P}_{\mathcal{M}(\underline{P})} = \underline{P}$. \square

Stelling 4.4

Als we aan de hand van een credale verzameling \mathcal{M} de onderprevisie $\underline{P}_{\mathcal{M}}$ invoeren via vergelijking (4.2), dan is $\mathcal{M}(\underline{P}_{\mathcal{M}}) = \mathcal{M}$.

Bewijs

Merk om te beginnen op dat $\mathcal{M} \subseteq \mathcal{M}(\underline{P}_{\mathcal{M}})$. Dit kan makkelijk ingezien worden aan de hand van definitie (4.2) en (4.6). We moeten dus enkel nog aantonen dat $\mathcal{M}(\underline{P}_{\mathcal{M}}) \subseteq \mathcal{M}$,

waarvoor we een bewijs uit het ongerijmde geven.

Stel dat er een $q \in \mathcal{M}(\underline{P}_{\mathcal{M}})$ bestaat die geen element is van \mathcal{M} . Aangezien \mathcal{M} bij onderstelling een credale verzameling is, en dus gesloten en convex is, kunnen we gebruik maken van een versie van de scheidende hypervlak-stelling voor eindige ruimten [7, Corollary 11.4.2]. We kunnen uit deze stelling afleiden dat er een $f \in \mathcal{L}(\mathcal{X})$ en een $\alpha \in \mathbb{R}$ bestaan waarvoor $P_q(f) = \sum_{z \in \mathcal{X}} q(z)f(z) < \alpha$ en $P_p(f) = \sum_{z \in \mathcal{X}} p(z)f(z) > \alpha$ voor alle $p \in \mathcal{M}$. De onderstelling dat $q \in \mathcal{M}(\underline{P}_{\mathcal{M}})$ impliceert echter op zijn beurt dat $P_q(f) \geq \underline{P}_{\mathcal{M}}(f) = \min \{P_p(f) : p \in \mathcal{M}\}$. Dit is een contradictie, en dus kunnen we besluiten dat er geen $q \in \mathcal{M}(\underline{P}_{\mathcal{M}})$ bestaat die niet een element is van \mathcal{M} en dus dat $\mathcal{M}(\underline{P}_{\mathcal{M}}) \subseteq \mathcal{M}$. \square

Uit de bovenstaande stellingen volgt dat er een één–één-verband bestaat tussen de verzameling van coherente onderprevisies \underline{P} en de ermee corresponderende credale verzamelingen $\mathcal{M}(\underline{P})$. Redeneren met coherente onderprevisies leidt dus tot dezelfde conclusies als redeneren met credale verzamelingen. Hoewel er behalve onderprevisies ook andere methoden zijn om te redeneren met credale verzamelingen, zullen we in deze masterproef steeds met onderprevisies werken. We doen dit omdat onderprevisies over enkele eigenschappen beschikken die ons goed van pas zullen komen.

Tot slot wil ik u nog even de essentie van deze paragraaf meegeven, namelijk dat coherente onderprevisies niet zomaar een abstract wiskundig model zijn, maar dat ze een wiskundig handige manier zijn om robuust te redeneren met credale verzamelingen van massafuncties.

4.6 Conditionele onderprevisies

We noemen een voor alle $f \in \mathcal{L}(\mathcal{X})$ en $y \in \mathcal{Y}$ gedefinieerde functie $\underline{P}(\cdot|Y)$ een coherente conditionele onderprevisie als voor alle $y \in \mathcal{Y}$, de op $\mathcal{L}(\mathcal{X})$ gedefinieerde functie $\underline{P}(\cdot|y)$ een coherente onderprevisie is. Voor een vaste $f \in \mathcal{L}(\mathcal{X})$ kunnen we $\underline{P}(f|Y)$ ook interpreteren als een functie op \mathcal{Y} , die voor $Y = y$ de waarde $\underline{P}(f|y)$ aanneemt.

Aangezien voor alle $y \in \mathcal{Y}$ en alle $f \in \mathcal{L}(\mathcal{X} \times \mathcal{Y})$ de functie $f(\cdot, y)$ behoort tot $\mathcal{L}(\mathcal{X})$, is $\underline{P}(\cdot|y)$ bij uitbreiding ook gedefinieerd op $\mathcal{L}(\mathcal{X} \times \mathcal{Y})$. Het domein van de coherente conditionele onderprevisie $\underline{P}(\cdot|Y)$ kan dus eveneens uitgebreid worden tot $\mathcal{L}(\mathcal{X} \times \mathcal{Y})$. Voor elke $f \in \mathcal{L}(\mathcal{X} \times \mathcal{Y})$ en elke $y \in \mathcal{Y}$ definiëren we dus

$$\underline{P}(f|y) = \underline{P}(f(\cdot, y)|y).$$

Een dergelijke uitbreiding van het domein van een conditionele onderprevisie zal in het vervolg van deze masterproef steeds impliciet verondersteld worden.

4.7 Reguliere extensie

4.7.1 Een bevattelijke definitie

Een belangrijke vraag is nu hoe we een op $\mathcal{L}(\mathcal{X} \times \mathcal{Y})$ gedefinieerde onderprevisie \underline{P} kunnen conditioneren op de observatie $Y = y$. We beschouwen hiervoor de op $\mathcal{X} \times \mathcal{Y}$ gedefinieerde credale verzameling $\mathcal{M}(\underline{P})$ van massafuncties. Aangezien we in paragraaf 4.4 aantoonde dat redeneren met deze verzameling equivalent is met redeneren met de onderprevisie \underline{P} , worden we er toe geïnspireerd om de massafuncties $p \in \mathcal{M}(\underline{P})$ te conditioneren op de observatie $Y = y$. Als $p(y) > 0$ kunnen we dit doen met de regel van Bayes:

$$p(X|y) = \frac{p(X, y)}{p(y)}, \quad (4.7)$$

waarbij $p(y) = \sum_{x \in \mathcal{X}} p(x, y) = P_p(\mathbb{I}_{\{y\}})$. Als we dit toepassen voor alle $p \in \mathcal{M}(\underline{P})$ waarvoor $p(y) > 0$, dan bekomen we een nieuwe verzameling $\mathcal{M}_y(\underline{P})$ van massafuncties, die nu gedefinieerd zijn op \mathcal{X} .

$$p(X|y) \in \mathcal{M}_y(\underline{P}) \Leftrightarrow p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \quad (4.8)$$

Uit definitie (4.3) volgt dat deze verzameling niet leeg is als en slechts als $\bar{P}(\mathbb{I}_{\{y\}}) > 0$.

Voor elke coherente op $\mathcal{L}(\mathcal{X} \times \mathcal{Y})$ gedefinieerde onderprevisie \underline{P} , en alle $y \in \mathcal{Y}$ waarvoor $\bar{P}(\mathbb{I}_{\{y\}}) > 0$, voeren we nu op basis van de zogenaamde *reguliere extensie* [2, Appendix J], een op de gebeurtenis $Y = y$ geconditioneerde onderprevisie $\underline{P}(\cdot|y)$ in, die gedefinieerd is voor alle $f \in \mathcal{L}(\mathcal{X} \times \mathcal{Y})$.

$$\underline{P}(f|y) = \min \{P_p(f(\cdot, y)) : p \in \mathcal{M}_y(\underline{P})\} \quad (4.9)$$

4.7.2 Een handiger uitdrukking

Definitie (4.9) is zeer bevattelijk, aangezien ze door haar gelijkheid met definitie (4.2) een goed inzicht geeft in waarom we er voor kiezen om op deze manier onze onderprevisie te conditioneren. Een groot nadeel van deze uitdrukking is echter dat ze ons verplicht om te redeneren met verzamelingen van massafuncties, terwijl we ons oorspronkelijke, niet geconditioneerde model hebben uitgedrukt als een onderprevisie. We vormen daarom definitie (4.9) om¹ tot een uitdrukking die rechtstreeks gebruik maakt van de onderprevisie \underline{P} : voor elke $f \in \mathcal{L}(\mathcal{X})$ geldt dat

$$\begin{aligned} \underline{P}(f|y) &= \min \{P_p(f(\cdot, y)) : p \in \mathcal{M}_y(\underline{P})\} \\ &= \min \left\{ \sum_{z \in \mathcal{X}} f(z, y) p(z) : p \in \mathcal{M}_y(\underline{P}) \right\} \\ &= \min \left\{ \sum_{z \in \mathcal{X}} f(z, y) p(z|y) : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\}. \end{aligned}$$

1. Deze omvorming wordt in zeer summere vorm beschreven in [2, Appendix J].

Aangezien we enkel $p \in \mathcal{M}(\underline{P})$ beschouwen waarvoor $p(y) > 0$ en we bovendien voor alle $(z, w) \in \mathcal{X} \times \mathcal{Y}$ weten dat $\mathbb{I}_{\{y\}}(z, w) = 1$ als $w = y$ en $\mathbb{I}_{\{y\}}(z, w) = 0$ als $w \neq y$, geldt dat

$$\begin{aligned} \sum_{z \in \mathcal{X}} f(z, y) p(z|y) &= \sum_{z \in \mathcal{X}} f(z, y) \frac{p(x, y)}{p(y)} \\ &= \left(\sum_{(z, w) \in \mathcal{X} \times \mathcal{Y}} \mathbb{I}_{\{y\}}(z, w) f(z, y) p(x, y) \right) \frac{1}{p(y)} \\ &= P_p(\mathbb{I}_{\{y\}} f) \frac{1}{p(y)} = \frac{P_p(\mathbb{I}_{\{y\}} f)}{P_p(\mathbb{I}_{\{y\}})}, \end{aligned}$$

en dus dat

$$\underline{P}(f|y) = \min \left\{ \frac{P_p(\mathbb{I}_{\{y\}} f)}{P_p(\mathbb{I}_{\{y\}})} : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\}.$$

Doordat voor elke $\alpha \in \mathbb{R}$, $\alpha = \max \{ \mu \in \mathbb{R} : \alpha \geq \mu \}$, kunnen we de bovenstaande uitdrukking ook schrijven als

$$\underline{P}(f|y) = \max \left\{ \mu \in \mathbb{R} : \min \left\{ \frac{P_p(\mathbb{I}_{\{y\}} f)}{P_p(\mathbb{I}_{\{y\}})} : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\} \geq \mu \right\}. \quad (4.10)$$

De hierin voorkomende ongelijkheid kunnen we nu als volgt verder omvormen:

$$\begin{aligned} &\min \left\{ \frac{P_p(\mathbb{I}_{\{y\}} f)}{P_p(\mathbb{I}_{\{y\}})} : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\} \geq \mu \\ \Leftrightarrow &\min \left\{ \frac{P_p(\mathbb{I}_{\{y\}} f)}{P_p(\mathbb{I}_{\{y\}})} - \mu : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\} \geq 0 \\ \Leftrightarrow &\min \left\{ \frac{P_p(\mathbb{I}_{\{y\}} [f - \mu])}{p(y)} : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \right\} \geq 0 \\ \Leftrightarrow &\min \{ P_p(\mathbb{I}_{\{y\}} [f - \mu]) : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \} \geq 0. \end{aligned} \quad (4.11)$$

Hierbij volgt de tweede equivalentie uit de homogeniteit en lineariteit van de previsie P_p en de laatste equivalentie uit de voorwaarde $p(y) > 0$.

Als $p(y) = 0$, dan is voor alle $x \in \mathcal{X}$ ook $p(x, y) = 0$ en dus is

$$P_p(\mathbb{I}_{\{y\}} [f - \mu]) = \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{Y}} p(x, z) \mathbb{I}_{\{z\}} [f(x, z) - \mu] = \sum_{x \in \mathcal{X}} p(x, y) \mathbb{I}_{\{z\}} [f(x, y) - \mu] = 0.$$

Hieruit volgt dat we de voorwaarde $p(y) > 0$ kunnen laten vallen in ongelijkheid (4.11):

$$\begin{aligned} &\min \{ P_p(\mathbb{I}_{\{y\}} [f - \mu]) : p \in \mathcal{M}(\underline{P}) \text{ en } p(y) > 0 \} \geq 0 \\ \Leftrightarrow &\min \{ P_p(\mathbb{I}_{\{y\}} [f - \mu]) : p \in \mathcal{M}(\underline{P}) \} \geq 0 \\ \Leftrightarrow &\underline{P}(\mathbb{I}_{\{y\}} [f - \mu]) \geq 0. \end{aligned}$$

Als we de oorspronkelijke ongelijkheid in uitdrukking (4.10) hierdoor vervangen, geeft dit tot slot de vorm waarin reguliere extensie het meest gekend is [2] en waarin ze ook het handigst kan toegepast worden, aangezien ze rechtstreeks de onderprevisie \underline{P} gebruikt en niet de omweg via een verzameling van massafuncties maakt:

$$\underline{P}(f|y) = \max \{ \mu \in \mathbb{R} : \underline{P}(\mathbb{I}_{\{y\}}[f - \mu]) \geq 0 \}. \quad (4.12)$$

4.7.3 Enkele eigenschappen

In de onderstaande eigenschappen en hun bewijzen (en ook in het vervolg van deze masterproef) wordt veelvuldig gebruik gemaakt van begrippen zoals positief, negatief, dalend en stijgend. We verduidelijken daarom eerst even kort wat we met deze begrippen bedoelen.

We noemen een getal positief als het strikt groter is dan nul en negatief als het strikt kleiner is dan nul. Een getal dat zowel groter dan nul kan zijn als gelijk kan zijn aan nul, zullen we dus niet “positief” noemen. We spreken in dat geval van een niet-negatief getal. Analooft spreken we ook over een niet-positief getal.

$$\begin{aligned} x > 0 &\Leftrightarrow x \text{ is positief} & x \geq 0 &\Leftrightarrow x \text{ is niet-negatief} \\ x < 0 &\Leftrightarrow x \text{ is negatief} & x \leq 0 &\Leftrightarrow x \text{ is niet-positief} \end{aligned}$$

Voor de begrippen stijgend en dalend gelden analoge uitspraken. We noemen een functie pas stijgend als ze strikt stijgt en dalend als ze strikt daalt. Als een functie stijgt, maar niet strikt stijgt, dan noemen we ze niet-dalend en als ze daalt, maar niet strikt daalt, dan noemen we ze niet-stijgend. Voor een functie f , gedefinieerd op \mathbb{R} , zeggen we

$$\begin{aligned} (x > y \Leftrightarrow f(x) > f(y)) &\Leftrightarrow f \text{ is stijgend} \\ (x > y \Leftrightarrow f(x) < f(y)) &\Leftrightarrow f \text{ is dalend} \\ (x > y \Leftrightarrow f(x) \geq f(y)) &\Leftrightarrow f \text{ is niet-dalend} \\ (x > y \Leftrightarrow f(x) \leq f(y)) &\Leftrightarrow f \text{ is niet-stijgend,} \end{aligned}$$

waarbij x en y twee willekeurige elementen van \mathbb{R} zijn.

De in definitie (4.12) voorkomende functie $\rho_f(\mu) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu])$ vertoont door de coherentie van \underline{P} een aantal nuttige eigenschappen. Ze zullen ons later goed van pas komen om conclusies te trekken over de conditionele previsie $\underline{P}(f|y)$. De onderstaande eigenschappen worden bewezen in Bijlage A.

Eigenschap 4.5

$\rho_f(\mu)$ is een niet-stijgende, concave en (Lipschitz-)continue functie van μ .

Eigenschap 4.6

Er bestaat een $\mu^- < 0$ waarvoor $\rho_f(\mu^-) \geq 0$ en een $\mu^+ > 0$ waarvoor $\rho_f(\mu^+) \leq 0$. Wegens de continuïteit van ρ_f (Eigenschap 4.5) impliceert dit tevens dat ρ_f minstens 1 nulpunt bezit.

Eigenschap 4.7

Als $\underline{P}(\{y\}) > 0$, dan is $\rho_f(\mu)$ een dalende functie van μ met een uniek nulpunt.

Eigenschap 4.8

Als $\bar{P}(\{y\}) = 0$, dan is $\rho_f(\mu)$ identisch gelijk aan nul over zijn hele domein.

Eigenschap 4.9

Als $\underline{P}(\{y\}) = 0$ en $\bar{P}(\{y\}) > 0$ dan is $\rho_f(\mu)$ in het interval $]-\infty, \underline{P}(f|y)]$ gelijk aan nul en in het interval $]\underline{P}(f|y), +\infty[$ negatief en dalend.

Eigenschap 4.10

Als er een a bestaat waarvoor $\rho_f(a) > 0$ dan is $\rho_f(\mu)$ een dalende functie van μ .

Eigenschap 4.11

Als $\rho_f(\mu)$ in een willekeurig interval $]a, b[$ negatief is, dan is $\rho_f(\mu)$ in ditzelfde interval ook dalend.

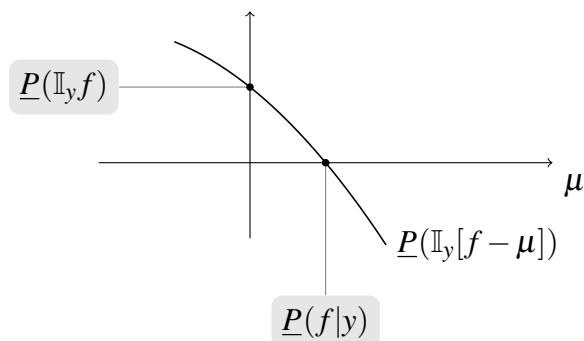
Uit deze eigenschappen kunnen we bovendien de volgende nuttige stelling afleiden.

Stelling 4.5

Als de onderwaarschijnlijkheid $\underline{P}(\{y\})$ van de gebeurtenis waarop we conditioneren positief is, dan is voor elke functie f het teken van $\underline{P}(\mathbb{I}_y, f)$ gelijk aan dat van $\underline{P}(f|y)$: ze zijn beiden positief, beiden negatief of beiden gelijk aan nul.

Bewijs

$\rho_f(\mu)$ is in dit geval een continue (Eigenschap 4.5), dalende (Eigenschap 4.7) functie die een uniek nulpunt heeft. Uit vergelijking (4.12) volgt dat $\underline{P}(f|y)$ overeenstemt met dit unieke nulpunt.² Verder weten we dat $\underline{P}(\mathbb{I}_y, f)$ het eveneens unieke snijpunt is van $\rho_f(\mu)$ met de verticale $\mu = 0$. Het gestelde kan dan vervolgens makkelijk ingezien worden aan de hand van de onderstaande figuur.



□

2. Reguliere extensie is dan bovendien de enige coherente manier om te conditioneren [8, §3.2] en we spreken in dat geval van de veralgemeende regel van Bayes. Hij reduceert zich tot de gewone regel van Bayes als de beschouwde modellen precies zijn.

Imprecieze hidden Markovmodellen

“... all models are approximations. Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind ...”

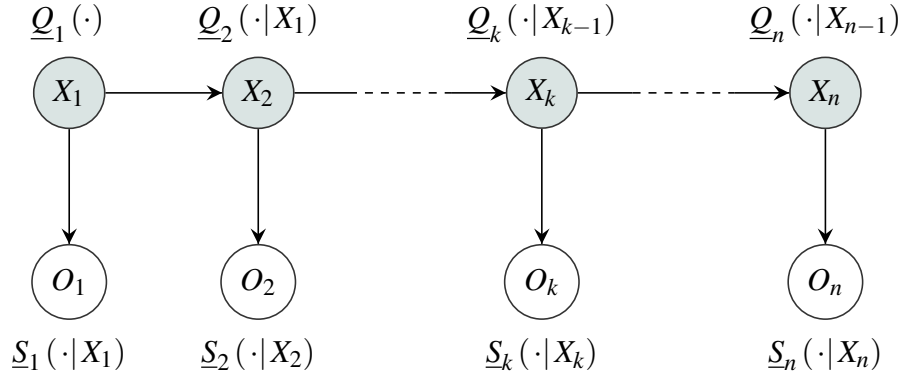
– George E. P. Box

5.1 Inleiding

In hoofdstuk 3 hebben we gezien wat een hidden Markovmodel is, hoe het gebruikt kan worden en waarom dit soms een handige manier is om de werkelijkheid te modelleren. Dit model was precies, en ging er dus van uit dat we voldoende informatie hebben, of voldoende zeker zijn van onze persoonlijke overtuigingen om de gebruikte lokale modellen exact te kunnen bepalen. In hoofdstuk 4 hebben we verteld dat het opgeven van een precies model lang niet altijd te verantwoorden is, en dat het soms veel beter kan zijn om de imprecisie in onze gegevens of onze overtuigingen toe te laten en op correcte wijze te modelleren. Een goede manier om dit te doen bleek het gebruik van coherente onderprevisies. U begrijpt ongetwijfeld wat de bedoeling van het huidige hoofdstuk zal zijn ...

5.2 Hetzelfde, maar “met streepjes eronder”

Als we de theorie van coherente onderprevisies toepassen op een hidden Markovmodel, dan bekomen we in essentie opnieuw een hidden Markovmodel, waarbij we nu echter ook in staat zijn om imprecieze gegevens of overtuigingen te modelleren. We spreken in dat geval van een *imprecies hidden Markovmodel* (iHMM). In dit hoofdstuk zullen we uitleggen hoe een imprecies hidden Markovmodel wordt opgebouwd, er recursieve betrekkingen voor opstellen en enkele handige eigenschappen afleiden. We vallen meteen met de deur in huis aan de hand van de onderstaande schematische weergave van een iHMM.



De aandachtige lezer merkt op dat dit model bijna identiek is aan het precieze hidden Markovmodel uit hoofdstuk 3, met dit verschil dat er deze keer “streepjes onder staan”. We zullen verderop merken dat deze simplistische beschrijving in vele gevallen stand houdt. Heel wat van de vergelijkingen die we voor iHMM’s zullen afleiden zijn namelijk nagenoeg hetzelfde als die voor HMM’s, maar dan met “streepjes eronder”. Een grote troef van de hier voorgestelde iHMM is dus dat hoewel in het algemeen de imprecieze waarschijnlijkheidsleer heel wat complexer is dan de precieze, in dit specifieke geval de te gebruiken uitdrukkingen toch relatief eenvoudig blijven.

5.3 Lokale modellen

De veranderlijken waaruit een iHMM bestaat, zijn identiek aan die van een HMM, namelijk een sequentie $X_{1:n}$ van verborgen toestanden en een sequentie $O_{1:n}$ van observaties. Voor elk van deze veranderlijken beschikken we over een lokaal model, in de vorm van een al dan niet conditionele onderprevisie, waarvan we eisen dat ze coherent is.

De eerste toestandsveranderlijke X_1 wordt gemodelleerd door het *marginaal* model \underline{Q}_1 , dat gedefinieerd is op $\mathcal{L}(\mathcal{X}_1)$. Voor elke toestandsveranderlijke X_k , met $k \in \{2, \dots, n\}$, hebben we een lokale conditionele onderprevisie $\underline{Q}_k(\cdot|X_{k-1})$, die geconditioneerd is op de vorige toestandsveranderlijke en gedefinieerd is op $\mathcal{L}(\mathcal{X}_k)$. We noemen dit *transitiemodellen*. Analooq aan de notatie voor een HMM, zullen we \underline{Q}_1 soms identificeren met de conditionele onderprevisie $\underline{Q}_1(\cdot|X_0)$, waarbij X_0 slechts één enkele waarde kan aannemen. Voor elke observatie O_k beschikken we tot slot over een conditionele lokale previsie $\underline{S}_k(\cdot|X_k)$. Deze *emissiemodellen* zijn geconditioneerd op hun corresponderende toestandsveranderlijke en gedefinieerd op $\mathcal{L}(\mathcal{O}_k)$.

Behalve coherentie, leggen we in het kader van deze masterproef nog een extra eis op aan de gebruikte lokale previsies.

Aanname 2

Alle lokale bovenwaarschijnlijkheden zijn positief. Hiermee bedoelen we dat voor alle $k \in \{1, \dots, n\}$, alle $x_k \in \mathcal{X}_k$, alle $x_{k-1} \in \mathcal{X}_{k-1}$ en alle $o_k \in \mathcal{O}_k$ zowel $\overline{Q}_k(\{x_k\}|x_{k-1})$ als $\overline{S}_k(\{o_k\}|x_k)$ strikt groter dan nul zijn.

Aan de hand van definitie (4.1) en (4.3) kunnen we makkelijk inzien dat deze aanname niet al te streng is. Als voor een op $\mathcal{L}(\mathcal{X})$ gedefinieerde bovenprevisie $\bar{P}_{\mathcal{M}}(\cdot)$ en een $x \in \mathcal{X}$ geldt dat $\bar{P}_{\mathcal{M}}(\mathbb{I}_{\{x\}}) = 0$, dan volgt hier namelijk uit dat

$$\begin{aligned}\bar{P}_{\mathcal{M}}(\mathbb{I}_{\{x\}}) &= \max \{P_p(\mathbb{I}_{\{x\}}) : p \in \mathcal{M}\} \\ &= \max \left\{ \sum_{z \in \mathcal{X}} \mathbb{I}_{\{x\}}(z)p(z) : p \in \mathcal{M} \right\} \\ &= \max \{p(x) : p \in \mathcal{M}\} = 0,\end{aligned}$$

of dus dat voor alle massafuncties $p \in \mathcal{M}$ de waarschijnlijkheid $p(x)$ gelijk is aan nul. De gebeurtenis $X = x$ kan dus volgens ons model niet optreden en we kunnen x net zo goed weglaten uit de toestandsverzameling \mathcal{X} .

5.4 Interpretatie van de grafische structuur

We noemen een veranderlijke Y epistemisch irrelevant [2] voor een andere veranderlijke X als het kennen van de waarde van Y onze overtuigingen over de waarde van X niet verandert. Meer formeel impliceert dit dat als we een coherente onderprevisie $\underline{P}(\cdot)$ hebben, gedefinieerd op $\mathcal{L}(\mathcal{X})$, en Y is epistemisch irrelevant voor X , dat we dan uit $\underline{P}(\cdot)$ een coherente, conditionele onderprevisie $\underline{P}(\cdot|Y)$ kunnen afleiden. $\underline{P}(\cdot|Y)$ wordt hierbij voor alle $y \in \mathcal{Y}$ en alle $f \in \mathcal{L}(\mathcal{X})$ gegeven door $\underline{P}(f|y) = \underline{P}(f)$. We noemen twee veranderlijken X en Y epistemisch onafhankelijk [2], als X epistemisch irrelevant is voor Y en Y ook epistemisch irrelevant is voor X .

De grafische structuur van een iHMM wordt nu als volgt geïnterpreteerd [8]: conditioneel op de moeder van een veranderlijke, zijn alle veranderlijken die niet de moeder of één van de nakomelingen van deze veranderlijke zijn, *epistemisch irrelevant* voor die veranderlijke. De iHMM wordt hiermee een eenvoudige voorbeeld van een credaal netwerk [8–10].

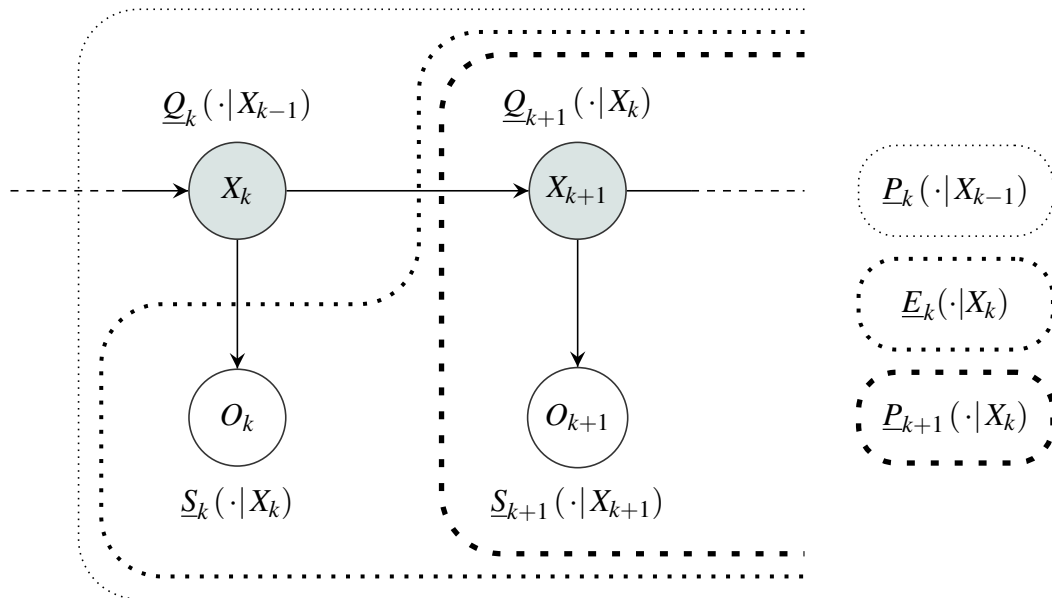
We bespreken kort één van de gevolgen van deze interpretatie [8]. Beschouw twee verschillende dochters van een zelfde moeder M . Verzameling A bevat één van deze dochters en alle nakomelingen van die dochter. Verzameling B bevat de andere dochter en al haar nakomelingen. Uit de interpretatie die we aan de grafische structuur hebben gegeven volgt dat, conditioneel op de moeder M , de elementen van A epistemisch irrelevant zijn voor de elementen van B en dat de elementen van B epistemisch irrelevant zijn voor de elementen van A . Conditioneel op de moeder M , is elk element van A dus epistemisch onafhankelijk van alle elementen uit B .

Het is nuttig om even de verschillen toe te lichten tussen de hier gebruikte grafische interpretatie en de interpretatie die we gebruikten voor een precieze HMM. Beide interpretaties zijn zeer analoog, met dit verschil dat we in een iHMM gebruik maken van epistemische irrelevantie en in een HMM van onafhankelijkheid.

Epistemische irrelevantie is een vorm van onafhankelijkheid voor imprecieze modellen. Men kan voor imprecieze modellen echter meerdere vormen van onafhankelijkheid invoeren. Naast epistemische irrelevantie kan men bijvoorbeeld ook gebruik maken van de eerder ingevoerde epistemische onafhankelijkheid of van de zogenaamde sterke onafhankelijkheid. In de literatuur wordt sterke onafhankelijkheid vaak gebruikt, maar in het kader van deze masterproef zullen we deze niet verder bespreken aangezien omdat we ze nooit invoeren.

Voor precieze modellen vallen de begrippen epistemische irrelevantie, epistemische onafhankelijkheid en sterke onafhankelijkheid allen samen en spreken we simpelweg van onafhankelijkheid [2]. Voor imprecieze modellen is epistemische irrelevantie echter een zwakkere uitdrukking dan sterke onafhankelijkheid, welke op zijn beurt zwakker is dan sterke onafhankelijkheid. Irrelevantie is bijvoorbeeld, in tegenstelling tot epistemische en sterke onafhankelijkheid, een asymmetrisch begrip. Het is net het gebruik van deze zwakkere notie van onafhankelijkheid die ons in staat stelt om de verderop in dit hoofdstuk besproken recursiebetrekkingen op te stellen. Meer informatie over de verschillende noties van onafhankelijkheid voor imprecieze waarschijnlijkheden is terug te vinden in [2, §9.1], [11] en [12].

5.5 Recursieve opbouw



Met de lokale modellen voor de toestandsveranderlijken en de observaties, willen we nu – net als bij de HMM – een globaal model opbouwen. Om dit recursief te kunnen aanpakken voeren we de conditionele onderprevisies $\underline{P}_k(\cdot | X_{k-1})$ in, die gedefinieerd zijn voor alle $f \in \mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$ en alle $k \in \{1, \dots, n\}$. De conditionele onderprevisie

$\underline{P}_1(\cdot|X_0)$ wordt hierbij (analoog aan het precieze geval) vereenzelvigd met het gezochte globale model $\underline{P}_1(\cdot)$.

Net als bij de HMM zijn we op zoek naar recursieve betrekkingen die ons in staat stellen om $\underline{P}_k(\cdot|X_{k-1})$ te bepalen aan de hand van $\underline{P}_{k+1}(\cdot|X_k)$. We gaan hiervoor op analoge wijze te werk.

Een eerste stap is het samenvoegen van het model $\underline{P}_{k+1}(\cdot|X_k)$ met het lokale model $\underline{S}_k(\cdot|X_k)$, welke respectievelijk gedefinieerd zijn op $\mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k+1:n})$ en $\mathcal{L}(\mathcal{O}_k)$. Deze twee modellen zijn beide coherente onderprevisies die conditioneel zijn op X_k en vormen samen het model $\underline{E}_k(\cdot|X_k)$, dat eveneens conditioneel is op X_k en gedefinieerd is op $\mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n})$. Merk op dat dit voor $k = n$ triviaal is omdat we dadelijk zien dat $\underline{E}_n(\cdot|X_n) = \underline{S}_n(\cdot|X_n)$. Voor $k \neq n$ is dit een minder voor de hand liggend probleem.

Er zijn zeer veel verschillende manieren om een gemeenschappelijk model $\underline{E}_k(\cdot|X_k)$ op te stellen, en we zullen dus bijkomende voorwaarden moeten opleggen. Een eerste voorwaarde is dat $\underline{E}_k(\cdot|X_k)$ op zichzelf ook een coherente onderprevisie moet zijn. Een tweede, voor de hand liggende voorwaarde is dat $\underline{E}_k(\cdot|X_k)$ moet samenvallen met $\underline{P}_{k+1}(\cdot|X_k)$ en $\underline{S}_k(\cdot|X_k)$ op hun respectievelijke domeinen. Een derde voorwaarde is dat het bekomen model moet voldoen aan alle irrelevantierelaties die we in paragraaf 5.4 hebben ingevoerd, meer bepaald dat het de epistemische onafhankelijkheid moet respecteren van enerzijds de veranderlijke O_k en anderzijds de veranderlijken X_{k+1}, \dots, X_n en O_{k+1}, \dots, O_n . De reden waarom deze veranderlijken epistemisch onafhankelijk zijn, werd reeds gegeven in paragraaf 5.4.

Een model dat aan al deze voorwaarden voldoet noemen we een onafhankelijk product van $\underline{P}_{k+1}(\cdot|X_k)$ en $\underline{S}_k(\cdot|X_k)$. Er bestaan echter nog steeds verschillende modellen die aan al deze voorwaarden voldoen. De puntsgewijs kleinste, of dus meest conservatieve van al deze onafhankelijke producten noemen we de *onafhankelijke natuurlijke extensie* van $\underline{P}_{k+1}(\cdot|X_k)$ en $\underline{S}_k(\cdot|X_k)$ en noteren we als $\underline{P}_{k+1}(\cdot|X_k) \otimes \underline{S}_k(\cdot|X_k)$ [11].

Samenvattend kunnen we de gezochte modellen $\underline{E}_k(\cdot|X_k)$ als volgt bepalen.

$$\underline{E}_k(\cdot|X_k) := \begin{cases} \underline{S}_n(\cdot|X_n) & k = n \\ \underline{S}_k(\cdot|X_k) \otimes \underline{P}_{k+1}(\cdot|X_k) & k = 1, \dots, n-1 \end{cases} \quad (5.1)$$

Het zou ons in het kader van deze masterproef te ver leiden om in detail in te gaan op het onafhankelijk product en de gebruikte onafhankelijke natuurlijke extensie. Wie meer informatie wenst kan ze terugvinden in [13] en [11]. Voor de doelstellingen van dit proefwerk volstaat het om te weten dat zulke onafhankelijke natuurlijke extensies

factoriserend zijn, wat impliceert dat

$$\begin{aligned}
\underline{E}_k(fg|z_k) &= \underline{E}_k(g\underline{E}_k(f|z_k)|z_k) \\
&= \underline{S}_k(g\underline{P}_{k+1}(f|z_k)|z_k) \\
&= \begin{cases} \underline{S}_k(g|z_k)\underline{P}_{k+1}(f|z_k) & \text{als } \underline{P}_{k+1}(f|z_k) \geq 0 \\ \overline{S}_k(g|z_k)\underline{P}_{k+1}(f|z_k) & \text{als } \underline{P}_{k+1}(f|z_k) \leq 0 \end{cases} \\
&= \overline{S}_k(g|z_k) \odot \underline{P}_{k+1}(f|z_k), \tag{5.2}
\end{aligned}$$

voor alle $z_k \in \mathcal{X}_k$, alle $f \in \mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k+1:n})$ en alle niet-negatieve $g \in \mathcal{L}(\mathcal{O}_k)$, met $k \in \{1, \dots, n-1\}$ (we noemen een functie niet-negatief als al haar functiewaarden dat zijn). De eerste gelijkheid is de hier bedoelde factorisatie-eigenschap. De tweede gelijkheid kunnen we schrijven omdat de onafhankelijke natuurlijke extensie van twee modellen op hun eigen domein samenvalt met die modellen. De derde gelijkheid volgt uit de toegevoegdheid (vergelijking (4.4)) en de positieve homogeniteit (coherentie-axioma C2) van $\underline{S}_k(\cdot|z_k)$ en voor de laatste gelijkheid maakten we gebruik van de handige verkorte notatie $\overline{m} \odot x = \underline{m} \max\{0, x\} + \overline{m} \min\{0, x\}$. We zullen de analoge notatie $\overline{m} \overline{n} \odot x = \underline{m} \underline{n} \max\{0, x\} + \overline{m} \overline{n} \min\{0, x\}$ verderop in deze masterproef soms ook gebruiken.

Een tweede en laatste stap in het recursief bepalen van $\underline{P}_k(\cdot|X_{k-1})$ is het samenvoegen van het model $\underline{E}_k(\cdot|X_k)$ met het lokale model $\underline{Q}_k(\cdot|X_{k-1})$. Deze modellen zijn respectievelijk gedefinieerd op $\mathcal{L}(\mathcal{X}_{k+1:n} \times \mathcal{O}_{k:n})$ en $\mathcal{L}(\mathcal{X}_k)$ en worden samengevoegd tot het op $\mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$ gedefinieerde model $\underline{P}_k(\cdot|X_{k-1})$. De meest conservatieve coherente manier om te doen is door gebruik te maken van marginale extensie [14]. Voor alle $k \in \{1, \dots, n\}$ wordt deze gegeven door

$$\underline{P}_k(\cdot|X_{k-1}) = \underline{Q}_k(\underline{E}_k(\cdot|X_k)|X_{k-1}). \tag{5.3}$$

Voor praktische berekeningen is het handig om in te zien dat dit equivalent is met zeggen dat voor alle $f \in \mathcal{L}(\mathcal{X}_{k:n} \times \mathcal{O}_{k:n})$

$$\underline{P}_k(f|X_{k-1}) = \underline{Q}_k\left(\sum_{z_k \in \mathcal{X}_k} \mathbb{I}_{\{z_k\}} \underline{E}_k(f(z_k, X_{k+1:n}, O_{k:n})|z_k) \Big| X_{k-1}\right).$$

5.6 Enkele handige eigenschappen

Met vergelijkingen (5.1)–(5.3) kunnen we nu vlot handige uitdrukkingen afleiden voor enkele interessante onder- en bovenwaarschijnlijkheden. Zo vinden we bijvoorbeeld

voor alle $k \in \{1, \dots, n\}$, $x_{k-1} \in \mathcal{X}_{k-1}$, $x_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$ dat

$$\begin{aligned} \underline{P}_k(\{x_{k:n}\} \times \{o_{k:n}\} | x_{k-1}) &= \underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | x_{k-1}) \\ &= \underline{Q}_k(\underline{E}_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | \mathbf{X}_k) | x_{k-1}) \\ &= \underline{Q}_k\left(\sum_{z_k \in \mathcal{X}_k} \mathbb{I}_{\{z_k\}} \underline{E}_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_k) | x_{k-1}\right) \\ &= \underline{Q}_k(\mathbb{I}_{\{x_k\}} \underline{E}_k(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k:n}\}} | x_k) | x_{k-1}). \end{aligned}$$

Door coherentie-axioma C1 weten we dat $\underline{E}_k(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k:n}\}} | x_k) \geq 0$, en kunnen we dus via coherentie-axioma C2 de bovenstaande vergelijking omvormen tot

$$= \underline{Q}_k(\mathbb{I}_{\{x_k\}} | x_{k-1}) \underline{E}_k(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k:n}\}} | x_k),$$

wat we op zijn beurt, omdat we opnieuw door coherentie-axioma C1 weten dat $\underline{P}_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | x_k) \geq 0$, met vergelijking (5.2) verder kunnen omvormen tot

$$\begin{aligned} &= \underline{Q}_k(\mathbb{I}_{\{x_k\}} | x_{k-1}) \underline{S}_k(\mathbb{I}_{\{o_k\}} | x_k) \underline{P}_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | x_k) \\ &= \underline{Q}_k(\{x_k\} | x_{k-1}) \underline{S}_k(\{o_k\} | x_k) \underline{P}_{k+1}(\mathbb{I}_{\{x_{k+1:n}\}} \mathbb{I}_{\{o_{k+1:n}\}} | x_k). \end{aligned}$$

Als we bovenstaande stappen steeds opnieuw uitvoeren, dan vinden we tot slot dat

$$\underline{P}_k(\{x_{k:n}\} \times \{o_{k:n}\} | x_{k-1}) = \prod_{i=k}^n \underline{Q}_i(\{x_i\} | x_{i-1}) \underline{S}_i(\{o_i\} | x_i), \quad (5.4)$$

Waarbij we in de allerlaatste stap, voor $k = n$, gebruik gemaakt hebben van de gelijkheid $\underline{E}_n(\{o_n\} | x_n) = \underline{S}_n(\{o_n\} | x_n)$ die rechtstreeks volgt uit vergelijking (5.1).

Op zeer analoge wijze kan men ook voor alle $k \in \{1, \dots, n\}$, $x_{k-1} \in \mathcal{X}_{k-1}$, $x_{k:n} \in \mathcal{X}_{k:n}$, $x_{k+1:n} \in \mathcal{X}_{k+1:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$ afleiden dat

$$\bar{P}_k(\{x_{k:n}\} \times \{o_{k:n}\} | x_{k-1}) = \prod_{i=k}^n \bar{S}_i(\{o_i\} | x_i) \bar{Q}_i(\{x_i\} | x_{i-1}), \quad (5.5)$$

dat

$$\underline{E}_k(\{x_{k+1:n}\} \times \{o_{k:n}\} | x_k) = \prod_{i=k+1}^n \underline{Q}_i(\{x_i\} | x_{i-1}) \prod_{i=k}^n \underline{S}_i(\{o_i\} | x_i) \quad (5.6)$$

$$\bar{E}_k(\{x_{k+1:n}\} \times \{o_{k:n}\} | x_k) = \prod_{i=k+1}^n \bar{Q}_i(\{x_i\} | x_{i-1}) \prod_{i=k}^n \bar{S}_i(\{o_i\} | x_i) \quad (5.7)$$

en dat

$$\underline{P}_k(\{x_{k:n}\} | x_{k-1}) = \prod_{i=k}^n \underline{Q}_i(\{x_i\} | x_{i-1}) \quad (5.8)$$

$$\bar{P}_k(\{x_{k:n}\} | x_{k-1}) = \prod_{i=k}^n \bar{Q}_i(\{x_i\} | x_{i-1}). \quad (5.9)$$

Hieruit volgen bovendien onderstaande nuttige eigenschappen, die in Bijlage A bewezen worden.

Eigenschap 5.1

Als alle lokale bovenwaarschijnlijkheden positief zijn, dan volgt hieruit voor alle $k \in \{1, \dots, n\}$, $x_{k-1} \in \mathcal{X}_{k-1}$, $x_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$ dat de bovenwaarschijnlijkheden $\bar{P}_k(\{x_{k:n}\}|x_{k-1})$, $\bar{P}_k(\{x_{k:n}\} \times \{o_{k:n}\}|x_{k-1})$, $\bar{P}_k(\{o_{k:n}\}|x_{k-1})$ en $\bar{E}_k(\{o_{k:n}\}|x_k)$ eveneens positief zijn. Wegens aanname 2 is dit in het kader van deze masterproef steeds het geval.

Eigenschap 5.2

Als alle lokale onderwaarschijnlijkheden positief zijn (hiermee bedoelen we dat voor alle $k \in \{1, \dots, n\}$, alle $x_k \in \mathcal{X}_k$, alle $x_{k-1} \in \mathcal{X}_{k-1}$ en alle $o_k \in \mathcal{O}_k$ zowel $\underline{Q}_k(\{x_k\}|x_{k-1})$ als $\underline{S}_k(\{o_k\}|x_k)$ strikt groter dan nul zijn), dan volgt hieruit dat voor alle $k \in \{1, \dots, n\}$, $x_{k-1} \in \mathcal{X}_{k-1}$, $x_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$ de onderwaarschijnlijkheden $\underline{P}_k(\{x_{k:n}\}|x_{k-1})$, $\underline{P}_k(\{x_{k:n}\} \times \{o_{k:n}\}|x_{k-1})$, $\underline{P}_k(\{o_{k:n}\}|x_{k-1})$ en $\underline{E}_k(\{o_{k:n}\}|x_k)$ eveneens positief zijn.

Hoofdstuk 6

Het EstiHMM-algoritme

“If we knew what it was we were doing, it would not be called research, would it?”

– Albert Einstein

6.1 Inleiding

Dit hoofdstuk bevat de kern van mijn onderzoek, namelijk de ontwikkeling van het EstiHMM¹-algoritme. Dit algoritme bepaalt op efficiënte wijze de maximale sequenties in een imprecies hidden Markovmodel. Het lost hiermee hetzelfde probleem op als het Viterbi-algoritme, maar behandelt dit probleem in een veel algemener geval, namelijk voor een imprecies model. Tot op heden bestond er voor zo'n imprecies model geen efficiënte oplossing voor dit probleem.

Dit algoritme is veruit de belangrijkste en meest originele bijdrage van deze masterproef. Alle resultaten die in dit hoofdstuk gepresenteerd worden zijn eigen werk. Als we gebruik maken van externe resultaten, dan wordt er expliciet gerefereerd.

In het eerste deel van dit hoofdstuk wordt een formele definitie opgebouwd voor de gezochte verzameling van maximale sequenties. We onderzoeken vervolgens hoe deze verzameling recursief opgebouwd kan worden en gebruiken de gevonden uitdrukkingen om het eerder opgestelde optimaliteitscriterium om te vormen tot een eenvoudige ongelijkheid.

In het tweede deel van dit hoofdstuk wordt vervolgens een algoritme ontwikkeld dat op efficiënte wijze de sequenties bepaalt die aan dit vereenvoudigde criterium voldoen. We bewijzen dat dit algoritme leidt tot een exacte oplossing en dat de complexiteit ervan, mits een goede implementatie, beperkt blijft.

1. Estimation in imprecise Hidden Markov Models

6.2 Conditioneren van het globaal model

De eerste stap in het bepalen van de maximale sequenties is het conditioneren van het globaal model van de iHMM op de waargenomen observaties. We maken hiervoor gebruik van de in paragraaf 4.7 ingevoerde reguliere extensie. Met behulp van deze reguliere extensie kunnen we de in hoofdstuk 5 opgebouwde modellen \underline{P}_k als volgt conditioneren. Voor elke $x_{k-1} \in \mathcal{X}_{k-1}$ geldt dat

$$\underline{P}_k(f|o_{k:n}, x_{k-1}) = \max \{ \mu \in \mathbb{R} : \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[f - \mu]|x_{k-1}) \geq 0 \} \quad (6.1)$$

Om de maximale sequenties van het imprecies hidden Markovmodel te bepalen zal het (zie paragraaf 6.3) zaak zijn om op efficiënte wijze het teken te achterhalen van deze conditionele globale onderprevisies. Meer bepaald voor een specifiek type functies $f \in \mathcal{L}(\mathcal{X}_{k:n})$ van de vorm $f = \mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}$.

Als alle lokale onderwaarschijnlijkheden positief zijn, dan volgt uit eigenschap 5.2 dat dit ook geldt voor de onderprevisies $\underline{P}_k(\{o_{k:n}\}|x_{k-1})$, voor alle $x_{k-1} \in \mathcal{X}_{k-1}$. Stelling 4.5 leert ons dan dat het teken van $\underline{P}_k(f|o_{k:n}, x_{k-1})$ gelijk is aan dat van $\underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}f|x_{k-1})$. Dit is een zeer handige eigenschap en het is één van de redenen waarom het later in dit hoofdstuk gepresenteerde algoritme zo efficiënt kan werken. Het komt in praktische toepassingen echter veelvuldig voor dat lokale onderwaarschijnlijkheden gelijk mogen zijn aan nul. In dat geval kunnen we stelling 4.5 niet meer toepassen en kunnen we voor een algemene functie f niet langer garanderen dat het teken van $\underline{P}_k(f|o_{k:n}, x_{k-1})$ gelijk is aan dat van $\underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}f|x_{k-1})$.

Om deze handige eigenschap toch te kunnen gebruiken, beperken we ons tot de specifieke functies van de vorm $f = \mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}$ en kijken we enkel naar onze recursief opgebouwde modellen $\underline{P}_k(\cdot|X_{k-1})$. In dat geval kunnen we aantonen dat stelling 4.5 geldig blijft onder zwakkere voorwaarden. We kunnen namelijk toelaten dat de lokale onderwaarschijnlijkheden nul worden, en eisen enkel dat de lokale bovenwaarschijnlijkheden positief zijn. Dit is een belangrijk resultaat dat later in dit hoofdstuk een cruciaal hulpmiddel zal blijken.

Stelling 6.1

Als alle lokale bovenwaarschijnlijkheden positief zijn,² dan is voor elke $k \in \{1, \dots, n\}$ en voor alle $x_{k-1} \in \mathcal{X}_{k-1}$ het teken van $\underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}]|x_{k-1})$ gelijk aan dat van $\underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}|o_{k:n}, x_{k-1})$. Ze zijn allebei positief, allebei negatief of allebei gelijk aan nul.

Bewijs

Fixeer eerst een willekeurige $x_{k-1} \in \mathcal{X}_{k-1}$. We beschouwen vervolgens de functie $\rho_k(\mu) = \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}} - \mu]|x_{k-1})$. Met behulp van vergelijking (6.1) zien we in dat $\underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}|o_{k:n}, x_{k-1})$ hiervan het meest rechtse nulpunt is. Bovendien

2. In het kader van deze masterproef wordt dit wegens aanname 2 steeds impliciet verondersteld.

weten we dat $\rho_k(0) = \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}]|x_{k-1})$. Uit eigenschap 4.5 leren we dat $\rho_k(\mu)$ niet-stijgend en continu is. Als $\rho_k(0) > 0$ heeft $\rho_k(\mu)$ dus (mede dankzij eigenschap 4.6) zeker een positief nulpunt en is $\underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}|o_{k:n}, x_{k-1}) > 0$. Als $\rho_k(0) < 0$, dan kan $\rho_k(\mu)$ enkel negatieve nulpunten hebben en weten we dus dat $\underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}|o_{k:n}, x_{k-1}) < 0$. Bijgevolg is het gestelde bewezen als we kunnen aantonen dat uit $\rho_k(0) = 0$ volgt dat $\rho_k(\varepsilon) < 0$ voor alle $\varepsilon > 0$, want dit impliceert dat $\underline{P}_k(\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}|o_{k:n}, x_{k-1}) = 0$. In het vervolg van dit bewijs zullen we aantonen dat dit inderdaad zo is. We beschouwen hiervoor twee afzonderlijke gevallen.

Het geval $x_k = \hat{x}_k$:

$$\begin{aligned} \rho_k(\varepsilon) &= \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}} - \varepsilon]|x_{k-1}) \\ &= \underline{Q}_k(\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}} - \varepsilon]|X_k)|x_{k-1}) \\ &= \underline{Q}_k\left(\mathbb{I}_{\{x_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon]|x_k) \right. \\ &\quad \left. + \sum_{z_k \neq x_k} \mathbb{I}_{\{z_k\}}\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}}|z_k)\Big| x_{k-1}\right) \end{aligned} \quad (6.2)$$

Van de coëfficiënten $\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}}|z_k)$ weten we dat ze gelijk zijn aan $-\varepsilon\bar{E}_k(\{o_{k:n}\}|z_k)$. Aangezien $\bar{E}_k(\{o_{k:n}\}|z_k)$ positief is door aanname 2 en eigenschap 5.1, volgt hieruit dat $\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}}|z_k)$ een negatieve, dalende functie van ε is.

Voor de coëfficiënt $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}}]|x_k)$ beschouwen we nu twee verschillende mogelijkheden.

Als $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}}]|x_k) > 0$, dan volgt uit eigenschap 4.10 onmiddellijk dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon]|x_k)$ een dalende functie van ε is. Het argument van $\underline{Q}_k(\cdot|X_{k-1})$ in vergelijking (6.2) is dus puntsgewijs dalend in ε , wat op zijn beurt door eigenschap 4.2 impliceert dat $\rho_k(\varepsilon)$ een dalende functie van ε is, en dus dat $\rho_k(\varepsilon) < \rho_k(0) = 0$.

Als daarentegen $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}}]|x_k) \leq 0$, dan kunnen we hieruit door eigenschap 4.3 afleiden dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon]|x_k) \leq 0$, en dus ook dat

$$\begin{aligned} \rho_k(\varepsilon) &\leq \underline{Q}_k\left(\sum_{z_k \neq x_k} \mathbb{I}_{\{z_k\}}\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}}|z_k)\Big| x_{k-1}\right) \\ &\leq \underline{Q}_k(\mathbb{I}_{\{z_{k*}\}}\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}}|z_{k*})\Big| x_{k-1}) \\ &= -\varepsilon\bar{E}_k(\{o_{k:n}\}|z_{k*})\bar{Q}_k(\{z_{k*}\}|x_{k-1}) < 0. \end{aligned}$$

Hierbij is z_{k*} een willekeurig gekozen $z_k \neq x_k$. De eerste twee ongelijkheden volgen uit eigenschap 4.3 en de laatste ongelijkheid volgt uit aanname 2 en eigenschap 5.1. Ook in dit geval vinden we dus dat $\rho_k(\varepsilon) < 0$.

Het geval $x_k \neq \hat{x}_k$:

$$\begin{aligned}
\rho_k(\varepsilon) &= \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}} - \varepsilon] | x_{k-1}) \\
&= \underline{Q}_k(\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}} - \varepsilon] | X_k) | x_{k-1}) \\
&= \underline{Q}_k\left(\mathbb{I}_{\{x_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \varepsilon] | x_k) + \mathbb{I}_{\{\hat{x}_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[-\mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon] | \hat{x}_k)\right. \\
&\quad \left. + \sum_{z_k \neq x_k, \hat{x}_k} \mathbb{I}_{\{z_k\}}\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}} | z_k) \Big| x_{k-1}\right)
\end{aligned} \tag{6.3}$$

Van de coëfficiënten $\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}} | z_k)$ hebben we in het bewijs voor het geval $x_k = \hat{x}_k$ al aangetoond dat ze negatieve, dalende functies van ε zijn. Mede daardoor en door eigenschap 4.3 weten we ook dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[-\mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon] | \hat{x}_k) \leq \underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}} | \hat{x}_k) < 0$, wat vervolgens wegens eigenschap 4.11 impliceert dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[-\mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon] | \hat{x}_k)$ een dalende functie van ε is.

Tot slot rest ons nog de coëfficiënt $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \varepsilon] | x_k)$. We beschouwen hiervoor twee mogelijkheden.

Als $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{x_{k+1:n}\}} | x_k) > 0$ dan weten we dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \varepsilon] | x_k)$ door eigenschap 4.10 een strikt dalende functie van ε is. Het argument van $\underline{Q}_k(\cdot | x_{k-1})$ in vergelijking (6.3) is dus puntsgewijs dalend in ε , wat op zijn beurt door eigenschap 4.2 impliceert dat $\rho_k(\varepsilon)$ een dalende functie van ε is, en dus ook dat $\rho_k(\varepsilon) < \rho_k(0) = 0$.

Als daarentegen $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{x_{k+1:n}\}} | x_k) = 0$, dan kunnen we hieruit door eigenschap 4.3 afleiden dat $\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k+1:n}\}} - \varepsilon] | x_k) \leq \underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{x_{k+1:n}\}} | x_k) = 0$, en dus ook, opnieuw door eigenschap 4.3, dat

$$\begin{aligned}
\rho_k(\varepsilon) &\leq \underline{Q}_k(\mathbb{I}_{\{\hat{x}_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[-\mathbb{I}_{\{\hat{x}_{k+1:n}\}} - \varepsilon] | \hat{x}_k) | x_{k-1}) \\
&\leq \underline{Q}_k(\mathbb{I}_{\{\hat{x}_k\}}\underline{E}_k(-\varepsilon\mathbb{I}_{\{o_{k:n}\}} | \hat{x}_k) | x_{k-1}) \\
&= -\varepsilon\bar{E}_k(\{o_{k:n}\} | \hat{x}_k)\bar{Q}_k(\{\hat{x}_k\} | x_{k-1}) < 0.
\end{aligned}$$

Voor de laatste ongelijkheid maakten we gebruik van aanname 2 en eigenschap 5.1. Ook voor dit geval vinden we dus dat $\rho_k(\varepsilon) < 0$. \square

6.3 Maximale sequenties

Een volgende stap bestaat er nu in om het op de observaties geconditioneerde globale model $\underline{P}_1(\cdot | o_{1:n})$ te gebruiken om de beste toestandssequenties te bepalen.

In het precieze geval konden we zeggen dat een toestandssequentie $\hat{x}_{1:n}$ beter was dan een toestandssequentie $x_{1:n}$ als voor het conditionele precieze model $P_1(\cdot | o_{1:n})$ gold dat $P_1(\{\hat{x}_{1:n}\} | o_{1:n}) > P_1(\{x_{1:n}\} | o_{1:n})$. Deze voorwaarde is door de lineariteit van een precieze previsie equivalent met $P_1(\mathbb{I}_{\{\hat{x}_{1:n}\}} - \mathbb{I}_{\{x_{1:n}\}} | o_{1:n}) > 0$.

Het imprecieze model $\underline{P}_1(\cdot|o_{1:n})$ kan door definitie (4.2) gezien worden als een verzameling van precieze modellen. Er zijn verschillende manieren om voor zo'n imprecies model een voorkeursrelatie te definiëren. Ik kies er hier voor om voor het imprecieze model $\underline{P}_1(\cdot|o_{1:n})$ pas te zeggen dat een sequentie beter is dan een andere, als dit voor alle onderliggende precieze modellen $P_1(\cdot|o_{1:n})$ het geval is.³ Uit definitie (4.2) volgt dan onmiddellijk dat we deze strikte voorkeur \succ tussen twee toestandssequenties $\hat{x}_{1:n}$ en $x_{1:n}$ als volgt kunnen invoeren:

$$\hat{x}_{1:n} \succ x_{1:n} \Leftrightarrow \underline{P}_1(\mathbb{I}_{\{\hat{x}_{1:n}\}} - \mathbb{I}_{\{x_{1:n}\}} | o_{1:n}) > 0.$$

We zeggen dan dat $x_{1:n}$ gedomineerd wordt door $\hat{x}_{1:n}$. De binaire relatie \succ is door de coherentie van $\underline{P}_1(\cdot|o_{1:n})$ zowel transitief als irreflexief en dus is \succ een strikte partiële orde.

De optimale toestandssequenties zijn vervolgens diegene die door geen enkele andere gedomineerd worden. Men spreekt van *maximale* of ongedomineerde sequenties.

$$\begin{aligned} \hat{x}_{1:n} \in \text{opt}(\mathcal{X}_{1:n}|o_{1:n}) &\Leftrightarrow (\forall x_{1:n} \in \mathcal{X}_{1:n}) x_{1:n} \not\succeq \hat{x}_{1:n} \\ &\Leftrightarrow (\forall x_{1:n} \in \mathcal{X}_{1:n}) \underline{P}_1(\mathbb{I}_{\{x_{1:n}\}} - \mathbb{I}_{\{\hat{x}_{1:n}\}} | o_{1:n}) \leq 0 \\ &\Leftrightarrow (\forall x_{1:n} \in \mathcal{X}_{1:n}) \underline{P}_1(\mathbb{I}_{\{o_{1:n}\}}[\mathbb{I}_{\{x_{1:n}\}} - \mathbb{I}_{\{\hat{x}_{1:n}\}}]) \leq 0 \end{aligned} \quad (6.4)$$

Hierbij maken we voor de laatste equivalentie gebruik van de belangrijke stelling 6.1.

Het doel van het EstiHMM-algoritme is om de verzameling $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ van maximale sequenties op een efficiënte manier te bepalen.

We zullen in het vervolg van dit hoofdstuk merken dat het nuttig is om algemenere verzamelingen van maximale sequenties te beschouwen. Voor elke $k \in \{1, \dots, n\}$ en elke $z_{k-1} \in \mathcal{Z}_{k-1}$ leiden we makkelijk af dat de maximale sequenties die corresponderen met de modellen $\underline{P}_k(\cdot|z_{k-1}, o_{k:n})$ gegeven worden door:

$$\begin{aligned} \hat{x}_{k:n} \in \text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \\ \Leftrightarrow (\forall x_{k:n} \in \mathcal{X}_{k:n}) \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}] | z_{k-1}) \leq 0. \end{aligned} \quad (6.5)$$

Merk op dat de verzameling $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ waar we uiteindelijk naar op zoek zijn, ook geschreven kan worden als $\text{opt}(\mathcal{X}_{1:n}|z_0, o_{1:n})$.

Het is nuttig om in te zien dat de hier ingevoerde verzamelingen van optimale sequenties voor imprecieze modellen op een mooie manier in verband staan met de in paragraaf 3.4.1 ingevoerde optimale sequenties voor precieze modellen. Zoals uit hoofdstuk 4 blijkt is elk imprecies model $\underline{P}_k(\cdot|z_{k-1})$ een overkoepelend model voor een verzameling van precieze modellen $P_k(\cdot|z_{k-1})$. Beschouw nu een willekeurige optimale sequentie $\hat{x}_{k:n}$ van een dergelijk precies model. Uit definitie (3.7) volgt dan dat

$$(\forall x_{k:n} \in \mathcal{X}_{k:n}) P_k(\mathbb{I}_{\{\hat{x}_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}) \geq P_k(\mathbb{I}_{\{x_{k:n}\}} \mathbb{I}_{\{o_{k:n}\}} | z_{k-1}),$$

3. Meer informatie is terug te vinden in [2, §3.9.2] en [15, §3.2].

wat equivalent is met

$$(\forall x_{k:n} \in \mathcal{X}_{k:n}) P_k \left(\mathbb{I}_{\{o_{k:n}\}} [\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}] \mid z_{k-1} \right) \leq 0.$$

Hieruit leiden we op zijn beurt via definitie (4.1) af dat

$$(\forall x_{k:n} \in \mathcal{X}_{k:n}) \underline{P}_k (\mathbb{I}_{\{o_{k:n}\}} [\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}] \mid z_{k-1}) \leq 0.$$

Uit definitie (6.5) blijkt nu dat $\hat{x}_{k:n}$ niet alleen een optimale sequentie van het precieze model $P_k(\cdot \mid z_{k-1})$ is, maar eveneens een optimale sequentie van het imprecieze model $\underline{P}_k(\cdot \mid z_{k-1})$ is. We kunnen dus besluiten dat de optimale toestandssequenties van elk precies model $P_k(\cdot \mid z_{k-1})$ dat vervat zit in het imprecies model $\underline{P}_k(\cdot \mid z_{k-1})$ ook optimale sequenties zijn van dit imprecies model. Een direct gevolg hiervan is onder andere dat elk imprecies model minstens één optimale sequentie heeft en dat de verzameling $\text{opt}(\mathcal{X}_{k:n} \mid z_{k-1}, o_{k:n})$ dus nooit leeg kan zijn.

Tot slot vermelden we nog dat de hier ingevoerde notatie voor een verzameling van optimale sequenties dezelfde is als degene die we invoerden in de uiteenzetting van het Viterbi-algoritme in paragraaf 3.4. Hetzelfde geldt voor enkele andere notaties die we verderop in dit hoofdstuk zullen invoeren. Dit is niet verwonderlijk aangezien deze definities telkens samenvallen voor precieze modellen. We gaan er van uit dat de lezer uit de context kan afleiden welke definitie bedoeld wordt.

6.4 Nuttige recursiebetrekkingen

Uit de voorgaande paragrafen en uit definitie (6.5) in het bijzonder, volgt dat het nuttig is om voor alle $k \in \{1, \dots, n\}$ over een makkelijk hanteerbare uitdrukking voor $\underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}] \mid z_{k-1})$ te beschikken, waarbij we $\Delta[x_{k:n}, \hat{x}_{k:n}]$ invoeren als verkorte notatie voor de op $\mathcal{X}_{k:n} \times \mathcal{O}_{k:n}$ gedefinieerde gok $\mathbb{I}_{\{o_{k:n}\}} [\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}]$. We zullen voor het afleiden van deze uitdrukking verschillende gevallen beschouwen.

Als $\hat{x}_k = x_k$ dan volgt voor alle $k \in \{1, \dots, n-1\}$ dat

$$\begin{aligned} \Delta[x_{k:n}, \hat{x}_{k:n}] &= \mathbb{I}_{\{o_{k:n}\}} [\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}] \\ &= \mathbb{I}_{\{o_k\}} \mathbb{I}_{\{x_k\}} \mathbb{I}_{\{o_{k+1:n}\}} [\mathbb{I}_{\{x_{k+1:n}\}} - \mathbb{I}_{\{\hat{x}_{k+1:n}\}}] \\ &= \mathbb{I}_{\{o_k\}} \mathbb{I}_{\{x_k\}} \Delta[x_{k+1:n}, \hat{x}_{k+1:n}] \end{aligned}$$

en dus ook dat

$$\begin{aligned} \underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}] \mid z_{k-1}) &= \underline{Q}_k(\underline{E}_k(\mathbb{I}_{\{o_k\}} \mathbb{I}_{\{x_k\}} \Delta[x_{k+1:n}, \hat{x}_{k+1:n}] \mid X_k) \mid z_{k-1}) \\ &= \underline{Q}_k(\mathbb{I}_{\{x_k\}} \underline{E}_k(\mathbb{I}_{\{o_k\}} \Delta[x_{k+1:n}, \hat{x}_{k+1:n}] \mid x_k) \mid z_{k-1}) \\ &= \underline{Q}_k(\{x_k\} \mid z_{k-1}) \odot \underline{E}_k(\mathbb{I}_{\{o_k\}} \Delta[x_{k+1:n}, \hat{x}_{k+1:n}] \mid x_k) \\ &= \underline{Q}_k(\{x_k\} \mid z_{k-1}) \underline{S}_k(\{o_k\} \mid x_k) \odot \underline{P}_{k+1}(\Delta[x_{k+1:n}, \hat{x}_{k+1:n}] \mid x_k). \quad (6.6) \end{aligned}$$

De eerste gelijkheid in deze afleiding is een rechtstreekse toepassing van vergelijking (5.3). De tweede gelijkheid kunnen we schrijven omdat voor alle $X_k \neq x_k$ geldt dat $\mathbb{I}_{\{x_k\}}(X_k) = 0$ en dus ook dat $\underline{E}_k(\mathbb{I}_{\{o_k\}}\mathbb{I}_{\{x_k\}}\Delta[x_{k+1:n}, \hat{x}_{k+1:n}]|x_k) = 0$. De derde gelijkheid volgt uit de toegevoegdheid (vergelijking (4.4)) en de positieve homogeniteit (coherentie-axioma C2) van $\underline{Q}_k(\cdot|z_{k-1})$ en de laatste gelijkheid volgt uit vergelijking (5.2).

Als $\hat{x}_n = x_n$ dan vinden we onmiddellijk dat

$$\begin{aligned} \underline{P}_n(\Delta[x_n, \hat{x}_n]|z_{n-1}) &= \underline{Q}_n(\underline{E}_k(\mathbb{I}_{\{o_n\}}[\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{\hat{x}_n\}}]|X_k)|z_{n-1}) \\ &= \underline{Q}_n(\underline{E}_k(\mathbb{I}_{\{o_n\}}[\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{x_n\}}]|X_k)|z_{n-1}) = 0. \end{aligned} \quad (6.7)$$

Als $\hat{x}_k \neq x_k$ dan kunnen we voor alle $k \in \{1, \dots, n\}$ afleiden dat

$$\begin{aligned} \underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) &= \underline{Q}_k(\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}[\mathbb{I}_{\{x_{k:n}\}} - \mathbb{I}_{\{\hat{x}_{k:n}\}}]|X_k)|z_{k-1}) \\ &= \underline{Q}_k(\mathbb{I}_{\{x_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{x_{k+1:n}\}}|x_k) + \mathbb{I}_{\{\hat{x}_k\}}\underline{E}_k(-\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{\hat{x}_{k+1:n}\}}|\hat{x}_k)|z_{k-1}) \\ &= \underline{Q}_k(\mathbb{I}_{\{x_k\}}\underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{x_{k+1:n}\}}|x_k) - \mathbb{I}_{\{\hat{x}_k\}}\bar{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{\hat{x}_{k+1:n}\}}|\hat{x}_k)|z_{k-1}) \\ &= \underline{Q}_k(\mathbb{I}_{\{x_k\}}\beta_k(x_{k:n}) - \mathbb{I}_{\{\hat{x}_k\}}\alpha_k(\hat{x}_{k:n})|z_{k-1}), \end{aligned} \quad (6.8)$$

waarbij voor elke $z_{k:n} \in \mathcal{X}_{k:n}$

$$\beta_k(z_{k:n}) = \underline{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{z_{k+1:n}\}}|z_k) = \underline{S}_k(\{o_k\}|z_k) \prod_{i=k+1}^n \underline{S}_i(\{o_i\}|z_i) \underline{Q}_i(\{z_i\}|z_{i-1}) \quad (6.9)$$

$$\alpha_k(z_{k:n}) = \bar{E}_k(\mathbb{I}_{\{o_{k:n}\}}\mathbb{I}_{\{z_{k+1:n}\}}|z_k) = \bar{S}_k(\{o_k\}|z_k) \prod_{i=k+1}^n \bar{S}_i(\{o_i\}|z_i) \bar{Q}_i(\{z_i\}|z_{i-1}). \quad (6.10)$$

De tweede gelijkheid volgt voor definities (6.9) en (6.10) respectievelijk uit vergelijkingen (5.6) en (5.7).

Voor elke toestandssequentie $z_{k:n} \in \mathcal{X}_{k:n}$, kunnen we $\alpha_k(z_{k:n})$ en $\beta_k(z_{k:n})$ makkelijk recursief bepalen:

$$\alpha_k(z_{k:n}) = \alpha_{k+1}(z_{k+1:n})\bar{S}_k(\{o_k\}|z_k)\bar{Q}_{k+1}(\{z_{k+1}\}|z_k) \quad (6.11)$$

$$\beta_k(z_{k:n}) = \beta_{k+1}(z_{k+1:n})\underline{S}_k(\{o_k\}|z_k)\underline{Q}_{k+1}(\{z_{k+1}\}|z_k), \quad (6.12)$$

voor $k \in \{1, \dots, n-1\}$, en beginnend bij:

$$\alpha_n(z_{n:n}) = \alpha_n(z_n) = \bar{S}_n(\{o_n\}|z_n)$$

$$\beta_n(z_{n:n}) = \beta_n(z_n) = \underline{S}_n(\{o_n\}|z_n).$$

6.5 Optimaliteitsprincipe

Hoewel definitie (6.5) relatief eenvoudig is, ligt het niet voor de hand om hiermee rechtstreeks de optimale sequenties $\hat{x}_{1:n}$ te vinden. Het aantal mogelijke sequenties

$x_{k:n} \in \mathcal{X}_{k:n}$ is namelijk exponentieel in de lengte van deze sequenties, waardoor de benodigde rekentijd zeer lang kan worden. We gaan daarom op zoek naar een elegantere methode om de optimale sequenties te bepalen, die gebruik maakt van recursie. De onderstaande stelling is een hulpmiddel om dit te kunnen doen.

Stelling 6.2 (Optimaliteitsprincipe)

Kies $k \in \{1, \dots, n-1\}$, $z_{k-1} \in \mathcal{X}_{k-1}$, $\hat{x}_{k:n} \in \mathcal{X}_{k:n}$ en $o_{k:n} \in \mathcal{O}_{k:n}$. Als $\underline{Q}_k(\{\hat{x}_k\}|z_{k-1}) > 0$ en $\underline{S}_k(\{o_k\}|\hat{x}_k) > 0$ dan geldt er dat:

$$\hat{x}_{k:n} \in \text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \Rightarrow \hat{x}_{k+1:n} \in \text{opt}(\mathcal{X}_{k+1:n}|\hat{x}_k, o_{k+1:n}).$$

Bewijs

Fixeer een $k \in \{1, \dots, n-1\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $\hat{x}_{k:n} \in \mathcal{X}_{k:n}$. We nemen aan dat $\hat{x}_{k+1:n}$ geen element is van $\text{opt}(\mathcal{X}_{k+1:n}|\hat{x}_k, o_{k+1:n})$ en tonen aan dat hieruit volgt dat $\hat{x}_{k:n}$ geen element is van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ als $\underline{Q}_k(\{\hat{x}_k\}|z_{k-1}) > 0$ en $\underline{S}_k(\{o_k\}|\hat{x}_k) > 0$.

Uit de veronderstelling en definitie (6.5) volgt dat er een $x_{k+1:n} \in \mathcal{X}_{k+1:n}$ is waarvoor $\underline{P}_{k+1}(\Delta[x_{k+1:n}, \hat{x}_{k+1:n}]|\hat{x}_k) > 0$. We verlengen deze sequentie $x_{k+1:n}$ door er vooraan de toestand \hat{x}_k aan toe te voegen, en krijgen zo de sequentie $x_{k:n}$, wat impliceert dat $\hat{x}_k = x_k$. Met vergelijking (6.6) vinden we dan dat

$$\underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) = \underline{Q}_k(\{\hat{x}_k\}|z_{k-1})\underline{S}_k(\{o_k\}|\hat{x}_k)\underline{P}_{k+1}(\Delta[x_{k+1:n}, \hat{x}_{k+1:n}]|\hat{x}_k) > 0,$$

waarmee bewezen is dat $\hat{x}_{k:n}$ inderdaad geen element is van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$. \square

Een direct gevolg van dit optimaliteitsprincipe is dat

$$\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \subseteq \text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}), \quad (6.13)$$

waarbij $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ de verzameling is van alle mogelijke optimale sequenties, die op grond van de bovenstaande redenering nog steeds een element kunnen zijn van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$:

$$\begin{aligned} & \text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \\ &= \left(\bigcup_{z_k \in \text{Pos}_k(z_{k-1})} z_k \oplus \text{opt}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n}) \right) \cup \left(\bigcup_{z_k \notin \text{Pos}_k(z_{k-1})} z_k \oplus \mathcal{X}_{k+1:n} \right). \end{aligned} \quad (6.14)$$

De verzameling $\text{Pos}_k(z_{k-1}) \in \mathcal{X}_k$ definiëren we hierbij als

$$z_k \in \text{Pos}_k(z_{k-1}) \Leftrightarrow \underline{Q}_k(\{z_k\}|z_{k-1}) > 0 \text{ en } \underline{S}_k(\{o_k\}|z_k) > 0. \quad (6.15)$$

Merk op dat vergelijking (6.14) zich vereenvoudigt tot

$$\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) = \bigcup_{z_k \in \mathcal{X}_k} z_k \oplus \text{opt}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$$

als alle lokale onderwaarschijnlijkheden positief zijn. In het kader van deze masterproef laten we echter – om niet in te boeten aan de algemeenheid van het algoritme – toe dat

lokale onderwaarschijnlijkheden gelijk zijn aan nul en eisen we enkel dat de bovenwaarschijnlijkheden positief zijn, waardoor we dus de complexere, maar algemenere, vergelijking (6.14) moeten gebruiken.

We voeren tot slot nog voor elke $k \in \{1, \dots, n\}$, $s \in \{k, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$, $x_{k:s} \in \mathcal{X}_{k:s}$ en $o_{k:n} \in \mathcal{O}_{k:n}$ de onderstaande notatie in:

$$\text{mog}_{x_{k:s}}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) = \{z_{k:n} \in \text{mog}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n}) : z_{k:s} = x_{k:s}\}. \quad (6.16)$$

6.6 Voorafgaande berekeningen

Vooraleer we het eigenlijke algoritme bespreken, voeren we een aantal parameters in die ons nuttig van pas zullen komen in het algoritme en die makkelijk vooraf kunnen berekend worden.

We definiëren eerst voor alle $k \in \{1, \dots, n\}$ en $x_k \in \mathcal{X}_k$:

$$\alpha_k^{\max}(x_k) = \max_{\substack{z_{k:n} \in \mathcal{Z}_{k:n} \\ z_k = x_k}} \alpha_k(z_{k:n}) \quad (6.17)$$

en

$$\beta_k^{\max}(x_k) = \max_{\substack{z_{k:n} \in \mathcal{Z}_{k:n} \\ z_k = x_k}} \beta_k(z_{k:n}). \quad (6.18)$$

Aan de hand van vergelijkingen (6.11) en (6.12) zien we in dat we deze parameters makkelijk recursief kunnen berekenen:⁴

$$\begin{aligned} \alpha_k^{\max}(x_k) &= \max_{z_{k+1} \in \mathcal{Z}_{k+1}} \alpha_{k+1}^{\max}(z_{k+1}) \bar{S}_k(\{o_k\} | x_k) \bar{Q}_{k+1}(\{z_{k+1}\} | x_k) \\ &= \bar{S}_k(\{o_k\} | x_k) \max_{z_{k+1} \in \mathcal{Z}_{k+1}} \alpha_{k+1}^{\max}(z_{k+1}) \bar{Q}_{k+1}(\{z_{k+1}\} | x_k) \end{aligned} \quad (6.19)$$

en

$$\begin{aligned} \beta_k^{\max}(x_k) &= \max_{z_{k+1} \in \mathcal{Z}_{k+1}} \beta_{k+1}^{\max}(z_{k+1}) \underline{S}_k(\{o_k\} | x_k) \underline{Q}_{k+1}(\{z_{k+1}\} | x_k) \\ &= \underline{S}_k(\{o_k\} | x_k) \max_{z_{k+1} \in \mathcal{Z}_{k+1}} \beta_{k+1}^{\max}(z_{k+1}) \underline{Q}_{k+1}(\{z_{k+1}\} | x_k), \end{aligned} \quad (6.20)$$

voor $k \in \{1, \dots, n-1\}$ en beginnend bij

$$\alpha_n^{\max}(x_n) = \alpha_n(x_n) = \bar{S}_n(\{o_n\} | x_n) \quad (6.21)$$

en

$$\beta_n^{\max}(x_n) = \beta_n(x_n) = \underline{S}_n(\{o_n\} | x_n). \quad (6.22)$$

4. Dit is een vorm van dynamisch programmeren [16].

Vervolgens voeren we voor alle $k \in \{1, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $x_k, \hat{x}_k \in \mathcal{X}_k$, met $x_k \neq \hat{x}_k$, de onderstaande drempelwaarden in:

$$\theta_k(x_k, \hat{x}_k | z_{k-1}) = \min \left\{ a \geq 0 : \underline{Q}_k(\mathbb{I}_{\{x_k\}} - a\mathbb{I}_{\{\hat{x}_k\}} | z_{k-1}) \leq 0 \right\}. \quad (6.23)$$

Deze drempelwaarden kunnen – in het algemene geval – niet berekend worden met louter en alleen de lokale boven- en onderwaarschijnlijkheden. Uit het vervolg van dit hoofdstuk zal echter blijken dat dit wel zo is voor alle andere parameters waarvan het algoritme gebruik maakt. In praktische toepassingen zijn de lokale boven- en onderwaarschijnlijkheden bovendien vaak de enige gegevens waarover men beschikt, en het zou dus nuttig zijn om het EstiHMM-algoritme in dat geval toch te kunnen uitvoeren.

De reden waarom de drempelwaarden niet vastliggen als we de boven- en onderwaarschijnlijkheden van de lokale modellen kennen, is de volgende: in tegenstelling tot bij de precieze modellen, volstaan de waarden van de boven- en onderwaarschijnlijkheden van de verschillende toestanden niet om het volledige imprecieze model vast te leggen. Er bestaan namelijk nog steeds oneindig veel imprecieze modellen die allemaal deze boven- en onderwaarschijnlijkheden opleveren, maar toch onderling verschillen. We lossen dit probleem op door uit al deze mogelijke modellen het meest conservatieve, coherente model te kiezen, namelijk datgene met de grootste imprecisie.

Het meest conservatieve coherente model dat overeenstemt met een toekenning van lokale boven- en onderwaarschijnlijkheden is steeds 2-monotoon [17, 18]. We gaan in het kader van deze masterproef niet verder in op de precieze betekenis hiervan, het volstaat om te weten dat dit voor alle $a \geq 0$, $k \in \{1, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $x_k, \hat{x}_k \in \mathcal{X}_k$ impliceert dat:

$$\underline{Q}_k(\mathbb{I}_{\{x_k\}} - a\mathbb{I}_{\{\hat{x}_k\}} | z_{k-1}) = \underline{Q}_k(\{x_k\} | z_{k-1}) - a\overline{Q}_k(\{\hat{x}_k\} | z_{k-1}).$$

Hieruit volgt via definitie (6.23) – en mede door aanname 2 – dat

$$\theta_k(x_k, \hat{x}_k | z_{k-1}) = \frac{\underline{Q}_k(\{x_k\} | z_{k-1})}{\overline{Q}_k(\{\hat{x}_k\} | z_{k-1})}. \quad (6.24)$$

Deze uitdrukking is de kleinst mogelijke waarde van de drempelwaarde $\theta_k(x_k, \hat{x}_k | z_{k-1})$ die correspondeert met de gegeven onder- en bovenwaarschijnlijkheden $\underline{Q}_k(\{x_k\} | z_{k-1})$ en $\overline{Q}_k(\{\hat{x}_k\} | z_{k-1})$. Aangezien ze correspondeert met het meest conservatieve model zal ze ook leiden tot de meest conservatieve gevolgtrekkingen en dus tot de grootste verzameling van maximale sequenties. Het toepassen van vergelijking (6.24) om de drempelwaarden te berekenen is dus steeds een veilige benadering van de werkelijkheid.

Tot slot definiëren we voor alle $k \in \{1, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $\hat{x}_k \in \mathcal{X}_k$ ook nog

$$\alpha_k^{\text{opt}}(\hat{x}_k | z_{k-1}) = \max_{\substack{x_k \in \mathcal{X}_k \\ x_k \neq \hat{x}_k}} \beta_k^{\text{max}}(x_k) \theta_k(x_k, \hat{x}_k | z_{k-1}). \quad (6.25)$$

6.7 Herformulering van de optimaliteitsvoorwaarde

We herformuleren de optimaliteitsvoorwaarde eerst voor het geval $k = n$.

Voor elke $z_{n-1} \in \mathcal{X}_{n-1}$ is $\text{opt}(\mathcal{X}_n | z_{n-1}, o_n)$ volgens definitie (6.5) de verzameling van die elementen \hat{x}_n in \mathcal{X}_n waarvoor

$$(\forall x_n \in \mathcal{X}_n) \underline{P}_n(\mathbb{I}_{\{o_n\}}[\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{\hat{x}_n\}}] | z_{n-1}) \leq 0.$$

Als $x_n = \hat{x}_n$, dan weten we door vergelijking (6.7) dat $\underline{P}_n(\mathbb{I}_{\{o_n\}}[\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{\hat{x}_n\}}] | z_{n-1}) = 0$, zodat de bovenstaande voorwaarde equivalent is met

$$(\forall x_n \in \mathcal{X}_n \setminus \{\hat{x}_n\}) \underline{P}_n(\mathbb{I}_{\{o_n\}}[\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{\hat{x}_n\}}] | z_{n-1}) \leq 0,$$

wat op zijn beurt wegens vergelijking (6.8) hetzelfde is als eisen dat

$$(\forall x_n \in \mathcal{X}_n \setminus \{\hat{x}_n\}) \underline{Q}_n(\mathbb{I}_{\{x_n\}}\beta_n(x_n) - \mathbb{I}_{\{\hat{x}_n\}}\alpha_n(\hat{x}_n) | z_{n-1}) \leq 0,$$

of dus, door vergelijking (6.22), dat

$$(\forall x_n \in \mathcal{X}_n \setminus \{\hat{x}_n\}) \underline{Q}_n(\mathbb{I}_{\{x_n\}}\beta_n^{\max}(x_n) - \mathbb{I}_{\{\hat{x}_n\}}\alpha_n(\hat{x}_n) | z_{n-1}) \leq 0,$$

Voor die x_n waarvoor $\beta_n^{\max}(x_n) > 0$ weten we bovendien dat

$$\begin{aligned} \underline{Q}_n(\mathbb{I}_{\{x_n\}}\beta_n^{\max}(x_n) - \mathbb{I}_{\{\hat{x}_n\}}\alpha_n(\hat{x}_n) | z_{n-1}) \leq 0 &\Leftrightarrow \underline{Q}_n(\mathbb{I}_{\{x_n\}} - \mathbb{I}_{\{\hat{x}_n\}} \frac{\alpha_n(\hat{x}_n)}{\beta_n^{\max}(x_n)} | z_{n-1}) \leq 0 \\ &\Leftrightarrow \frac{\alpha_n(\hat{x}_n)}{\beta_n^{\max}(x_n)} \geq \theta_n(x_n, \hat{x}_n | z_{n-1}) \\ &\Leftrightarrow \alpha_n(\hat{x}_n) \geq \beta_n^{\max}(x_n) \theta_n(x_n, \hat{x}_n | z_{n-1}), \end{aligned}$$

waarbij de eerste equivalentie volgt uit coherentie-axioma C2 en de tweede equivalentie volgt uit definitie (6.23) en de wetenschap dat $\alpha_n(\hat{x}_n)$ per definitie altijd positief is.

Als $\beta_n^{\max}(x_n) = 0$, dan weten we omdat $\alpha_n(\hat{x}_n)$ altijd positief is dat

$$\begin{aligned} \underline{Q}_n(\mathbb{I}_{\{x_n\}}\beta_n^{\max}(x_n) - \mathbb{I}_{\{\hat{x}_n\}}\alpha_n(\hat{x}_n) | z_{n-1}) \leq 0 &\Leftrightarrow \underline{Q}_n(-\mathbb{I}_{\{\hat{x}_n\}}\alpha_n(\hat{x}_n) | z_{n-1}) \leq 0 \\ &\Leftrightarrow -\alpha_n(\hat{x}_n) \overline{Q}_n(\{\hat{x}_n\} | z_{n-1}) \leq 0. \end{aligned}$$

Deze laatste voorwaarde is altijd voldaan door aanname 2 en is bijgevolg equivalent met de voorwaarde $\alpha_n(\hat{x}_n) \geq \beta_n^{\max}(x_n) \theta_n(x_n, \hat{x}_n | z_{n-1})$, omdat deze ook altijd voldaan is, doordat $\beta_n^{\max}(x_n) = 0$.

Uit het voorgaande kunnen we afleiden dat \hat{x}_n een element is van $\text{opt}(\mathcal{X}_n | z_{n-1}, o_n)$ als en slechts als

$$(\forall x_n \in \mathcal{X}_n \setminus \{\hat{x}_n\}) \alpha_n(\hat{x}_n) \geq \beta_n^{\max}(x_n) \theta_n(x_n, \hat{x}_n | z_{n-1}).$$

Deze voorwaarde kunnen we via vergelijking (6.25) vereenvoudigen tot de ongelijkheid $\alpha_n(\hat{x}_n) \geq \alpha_n^{\text{opt}}(\hat{x}_n|z_{n-1})$, waardoor we kunnen concluderen dat

$$\text{opt}(\mathcal{X}_n|z_{n-1}, o_n) = \{\hat{x}_n \in \mathcal{X}_n : \alpha_n(\hat{x}_n) \geq \alpha_n^{\text{opt}}(\hat{x}_n|z_{n-1})\}. \quad (6.26)$$

Nu het geval $k = n$ behandeld is, kunnen we de optimaliteitsvoorwaarde herformuleren voor een willekeurige $k \in \{1, \dots, n-1\}$.

We fixeren een vaste $z_{k-1} \in \mathcal{Z}_{k-1}$ en zijn op zoek naar $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$. Door het optimaliteitsprincipe (6.13) weten we dat we ons kunnen beperken tot de sequenties $\hat{x}_{k:n}$ die tot de verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ behoren. Om na te gaan of een dergelijke $\hat{x}_{k:n}$ een element is van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$, moeten we volgens definitie (6.5) controleren of $\underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) \leq 0$ voor elke $x_{k:n} \in \mathcal{X}_{k:n}$.

Voor die $x_{k:n}$ waarvoor $x_k = \hat{x}_k$ zijn er twee mogelijkheden: ofwel behoort x_k tot $\text{Pos}_k(z_{k-1})$, ofwel niet.

Als x_k een element is van de verzameling $\text{Pos}_k(z_{k-1})$, dan weten we door definitie (6.14) dat $\hat{x}_{k+1:n} \in \text{opt}(\mathcal{X}_{k+1:n}|x_k, o_{k+1:n})$, waaruit volgt dat $\underline{P}_{k+1}(\Delta[x_{k+1:n}, \hat{x}_{k+1:n}]|x_k) \leq 0$. Dit impliceert op zijn beurt door vergelijking (6.6) dat $\underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) \leq 0$.

Als x_k geen element is van $\text{Pos}_k(z_{k-1})$, dan is $\underline{Q}_k(\{z_k\}|z_{k-1}) = 0$ of $\underline{S}_k(\{o_k\}|z_k) = 0$, waaruit wegens vergelijking (6.6) eveneens volgt dat $\underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) \leq 0$.

Voor alle $x_{k:n}$ waarvoor $x_k = \hat{x}_k$ is bijgevolg aan de te controleren ongelijkheid voldaan, zodat nagaan of $\hat{x}_{k:n}$ een element is van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ zich herleidt tot het controleren of voor elke $x_k \neq \hat{x}_k$ voldaan is aan

$$(\forall x_{k+1:n} \in \mathcal{X}_{k+1:n}) \underline{P}_k(\Delta[x_{k:n}, \hat{x}_{k:n}]|z_{k-1}) \leq 0,$$

wat wegens vergelijking (6.8) hetzelfde is als eisen dat

$$(\forall x_{k+1:n} \in \mathcal{X}_{k+1:n}) \underline{Q}_k(\mathbb{I}_{\{x_k\}} \beta_k(x_{k:n}) - \mathbb{I}_{\{\hat{x}_k\}} \alpha_k(\hat{x}_{k:n})|z_{k-1}) \leq 0.$$

Op zijn beurt is dit door eigenschap 4.3 en definitie (6.18) equivalent met

$$\underline{Q}_k(\mathbb{I}_{\{x_k\}} \beta_k^{\max}(x_k) - \mathbb{I}_{\{\hat{x}_k\}} \alpha_k(\hat{x}_{k:n})|z_{k-1}) \leq 0,$$

waarvan we door een redenering die volkomen analoog is aan degene die we toepasten voor het geval $k = n$ makkelijk inzien dat dit op hetzelfde neerkomt als eisen dat

$$\alpha_k(\hat{x}_{k:n}) \geq \beta_k^{\max}(x_k) \theta_k(x_k, \hat{x}_k|z_{k-1}).$$

Aangezien de bovenstaande voorwaarde moet gelden voor alle $x_k \neq \hat{x}_k$ vinden we tot slot dat dit equivalent is met eisen dat $\alpha_k(\hat{x}_{k:n}) \geq \alpha_k^{\text{opt}}(\hat{x}_k|z_{k-1})$, wat ons tot de conclusie leidt dat

$$\begin{aligned} & \text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) \\ &= \left\{ \hat{x}_{k:n} \in \text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n}) : \alpha_k(\hat{x}_{k:n}) \geq \alpha_k^{\text{opt}}(\hat{x}_k|z_{k-1}) \right\}. \end{aligned} \quad (6.27)$$

6.8 Het algoritme

6.8.1 Een recursieve oplossing

Het doel van het algoritme is om de verzameling $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ te bepalen. We gaan hiervoor recursief te werk.

Voor $k = n$ kunnen we met vergelijking (6.26) voor elke $z_{n-1} \in \mathcal{X}_{n-1}$ de verzameling $\text{opt}(\mathcal{X}_n|z_{n-1}, o_n)$ rechtstreeks berekenen.

Vervolgens bepalen we voor k gaande van $n-1$ tot 1 en voor elke $z_{k-1} \in \mathcal{X}_{k-1}$ de verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ uitgaande van de in de vorige recursiestap voor elke $z_k \in \mathcal{X}_k$ gevonden verzameling $\text{mog}(\mathcal{X}_{k+1:n}|z_k, o_{k+1:n})$. We doen dit door eerst de verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ op te bouwen met definitie (6.14), waarna we vergelijking (6.27) kunnen toepassen om $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ te bepalen.

Nadat we deze stappen hebben uitgevoerd voor alle k gaande van $n-1$ tot 1 , bekomen we zo tot slot voor $k = 1$ de gezochte verzameling $\text{opt}(\mathcal{X}_{1:n}|o_{1:n}) = \text{opt}(\mathcal{X}_{1:n}|z_0, o_{1:n})$.

6.8.2 Een efficiënte methode voor de essentiële stap

Het knelpunt in de bovenstaande redenering is voorlopig dat het toepassen van vergelijking (6.27) nog steeds een rekentijd kan vergen die exponentieel toeneemt met de lengte van de sequentie. De verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ kan namelijk een exponentieel aantal elementen bevatten, in welk geval het zeer inefficiënt zou zijn om voor al deze elementen $z_{k:n}$ één voor één te controleren of $\alpha_k(z_{k:n}) \geq \alpha_k^{\text{opt}}(z_k|z_{k-1})$. In het vervolg van deze paragraaf zullen we een elegante methode bespreken om dit probleem op een efficiënte, recursieve manier aan te pakken. We geven een concreet stappenplan om een deelverzameling van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ te construeren en tonen vervolgens aan dat de op die manier geconstrueerde deelverzameling gelijk is aan de gezochte verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$.

Vooraleer we aan het stappenplan beginnen, voer ik eerst nog voor een willekeurige $k \in \{1, \dots, n\}$, $s \in \{k, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $z_{k:s} \in \mathcal{X}_{k:s}$ de parameter $\alpha_k^{\text{opt}}(z_{k:s}|z_{k-1})$ in. Als $s = k$ dan stellen we $\alpha_k^{\text{opt}}(z_{k:k}|z_{k-1})$ gelijk aan de eerder in vergelijking (6.25) gedefinieerde $\alpha_k^{\text{opt}}(z_k|z_{k-1})$. Voor $s \in \{k+1, \dots, n\}$ kan $\alpha_k^{\text{opt}}(z_{k:s}|z_{k-1})$ vervolgens recursief bepaald worden met de recursiebetrekking

$$\alpha_k^{\text{opt}}(z_{k:s}|z_{k-1}) = \frac{\alpha_k^{\text{opt}}(z_{k:s-1}|z_{k-1})}{\bar{S}_{s-1}(\{o_{s-1}\}|\bar{z}_{s-1})\bar{Q}_s(\{z_s\}|\bar{z}_{s-1})}. \quad (6.28)$$

Het stappenplan gaat nu als volgt:

Controleer voor elke $x_k \in \mathcal{X}_k$ of voldaan is aan de voorwaarde

$$\alpha_k^{\text{max}}(x_k) \geq \alpha_k^{\text{opt}}(x_k|z_{k-1}). \quad (6.29)$$

De $x_k \in \mathcal{X}_k$ waarvoor aan deze voorwaarde voldaan is zullen we in het vervolg van deze uiteenzetting noteren als \hat{x}_k .

Kies vervolgens een willekeurige \hat{x}_k en controleer voor elke $x_{k+1} \in \mathcal{X}_{k+1}$ waarvoor $\text{mog}_{\hat{x}_k \oplus x_{k+1}}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ een niet-lege verzameling is of voldaan is aan de voorwaarde

$$\alpha_{k+1}^{\max}(x_{k+1}) \geq \alpha_k^{\text{opt}}(\hat{x}_k \oplus x_{k+1} | z_{k-1}). \quad (6.30)$$

Merk op dat $\alpha_k^{\text{opt}}(\hat{x}_k \oplus x_{k+1} | z_{k-1})$ met vergelijking (6.28) makkelijk kan berekend worden omdat we $\alpha_k^{\text{opt}}(\hat{x}_k | z_{k-1})$ al kennen uit de vorige stap. We noteren die $x_{k+1} \in \mathcal{X}_{k+1}$ waarvoor aan voorwaarde (6.30) voldaan is als \hat{x}_{k+1} en concateneren deze met de toestand \hat{x}_k om zo de sequenties $\hat{x}_{k:k+1}$ te bekomen. We voeren deze stap uit voor elke in de vorige stap bekomen \hat{x}_k en bekomen zo een verzameling van sequenties $\hat{x}_{k:k+1}$.

In een volgende stap beschouwen we in eerste instantie een willekeurige $\hat{x}_{k:k+1}$ en controleren we voor elke $x_{k+2} \in \mathcal{X}_{k+2}$ waarvoor $\text{mog}_{\hat{x}_{k:k+1} \oplus x_{k+2}}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ een niet-lege verzameling is of voldaan is aan de voorwaarde

$$\alpha_{k+2}^{\max}(x_{k+2}) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:k+1} \oplus x_{k+2} | z_{k-1}). \quad (6.31)$$

Merk op dat we $\alpha_k^{\text{opt}}(\hat{x}_{k:k+1} \oplus x_{k+2} | z_{k-1})$ makkelijk kunnen berekenen met vergelijking (6.28) omdat we $\alpha_k^{\text{opt}}(\hat{x}_{k+1} | z_{k-1})$ al berekend hebben in de vorige stap. We noteren die $x_{k+2} \in \mathcal{X}_{k+2}$ waarvoor aan voorwaarde (6.31) voldaan is als \hat{x}_{k+2} en concateneren deze met de sequentie $\hat{x}_{k:k+1}$ om zo de sequenties $\hat{x}_{k:k+2}$ te bekomen. We voeren deze stap uit voor elke in de vorige stap bekomen sequentie $\hat{x}_{k:k+1}$ en bekomen zo een hele verzameling van sequenties $\hat{x}_{k:k+2}$.

Op die manier gaan we verder tot we uiteindelijk over een verzameling van sequenties $\hat{x}_{k:n-1}$ beschikken. Voor elke willekeurige $\hat{x}_{k:n-1}$ controleren we tot slot voor elke $x_n \in \mathcal{X}_n$ waarvoor $\text{mog}_{\hat{x}_{k:n-1} \oplus x_n}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ een niet-lege verzameling is of voldaan is aan de voorwaarde

$$\alpha_n^{\max}(x_n) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:n-1} \oplus x_n | z_{k-1}). \quad (6.32)$$

We kunnen $\alpha_k^{\text{opt}}(\hat{x}_{k:n-1} \oplus x_n | z_{k-1})$ makkelijk berekenen met vergelijking (6.28) omdat we $\alpha_k^{\text{opt}}(\hat{x}_{k:n-1} | z_{k-1})$ al berekend hebben in de vorige stap. We noteren die $x_n \in \mathcal{X}_n$ waarvoor aan deze voorwaarde voldaan is als \hat{x}_n en concateneren deze met de sequentie $\hat{x}_{k:n-1}$ om zo de sequenties $\hat{x}_{k:n}$ te bekomen. We voeren deze stap uit voor elke in de vorige stap bekomen sequentie $\hat{x}_{k:n-1}$ en bekomen zo een hele verzameling van sequenties $\hat{x}_{k:n}$.

We zullen nu bewijzen dat de volgens het bovenstaande stappenplan opgebouwde verzameling van sequenties $\hat{x}_{k:n}$ gelijk is aan de gezochte verzameling $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ door enerzijds aan te tonen dat elke sequentie $\hat{x}_{k:n}$ in de opgebouwde verzameling een element is van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ en anderzijds aan te tonen dat een sequentie $z_{k:n}$ die niet in de met dit stappenplan opgebouwde verzameling zit nooit een element van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$ kan zijn.

We tonen eerst aan dat elke sequentie $\hat{x}_{k:n}$ in de volgens het bovenstaande stappenplan opgebouwde verzameling een element is van $\text{opt}(\mathcal{X}_{k:n} | z_{k-1}, o_{k:n})$.

Uit de laatste stap van het stappenplan blijkt dat elke $\hat{x}_{k:n}$ van de gevonden verzameling een element van $\text{mog}_{\hat{x}_{k:n}}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ is, en dus door definitie (6.16) ook tot de verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ behoort. Anderzijds volgt uit diezelfde laatste stap dat $\alpha_n^{\max}(\hat{x}_n) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:n}|z_{k-1})$, wat door vergelijking (6.21) en het herhaaldelijk toepassen van de recursiebetrekkingen (6.28) en (6.11) equivalent is met de ongelijkheid $\alpha_k(\hat{x}_{k:n}) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:n}|z_{k-1})$. Het volgt nu uit vergelijking (6.27) dat $\hat{x}_{k:n}$ een element is van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$.

Tot slot tonen we aan dat een sequentie $z_{k:n}$ die niet in de met het bovenstaande stappenplan opgebouwde verzameling zit nooit een element van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ kan zijn.

Als een sequentie $z_{k:n}$ niet in de volgens het stappenplan opgebouwde verzameling zit, dan wil dit zeggen dat er een $s \in \{k, \dots, n\}$ bestaat waarvoor $\alpha_s^{\max}(z_s) < \alpha_k^{\text{opt}}(z_{k:s}|z_{k-1})$, waaruit via vergelijking (6.17) volgt dat $\alpha_s(z_s) < \alpha_k^{\text{opt}}(z_{k:s}|z_{k-1})$. Door vervolgens herhaaldelijk de recursiebetrekkingen (6.28) en (6.11) toe te passen vinden we dat $\alpha_k(z_{k:n}) < \alpha_k^{\text{opt}}(z_k|z_{k-1})$. Uit deze laatste gelijkheid en vergelijking (6.27) volgt tot slot dat $z_{k:n}$ geen element van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ kan zijn.

We kunnen dus besluiten dat de verzameling van sequenties $\hat{x}_{k:n}$ die we met het in deze paragraaf besproken stappenplan bekommen de verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ is waar we naar op zoek waren.

6.8.3 De complexiteit van de essentiële stap

We bewijzen in deze paragraaf dat als we voor een $k \in \{1, \dots, n-1\}$ en een $z_{k-1} \in \mathcal{X}_{k-1}$ de verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ bepalen met het stappenplan dat we hierboven hebben uiteengezet, dan het aantal bewerkingen dat we hiervoor moeten uitvoeren lineair is in het aantal elementen van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ – of dus lineair in het aantal maximale sequenties –, lineair is in de lengte van de maximale sequenties en lineair is in het aantal mogelijke toestandswaarden. Dit is zeer efficiënt aangezien elk algoritme, hoe goed het ook ontworpen is, ten minste lineair moet zijn in haar aantal elementen en in de lengte van de sequenties. Het enkel en alleen al weergeven van de maximale sequenties heeft bijvoorbeeld al die complexiteit.

Het is eerst belangrijk om in te zien dat het bovenstaande stappenplan bestaat uit een groot aantal zeer analoge kleine stapjes, namelijk het toevoegen van een \hat{x}_s aan een eerder reeds bepaalde sequentie $\hat{x}_{k:s-1}$.⁵

Om voor een eerder reeds bepaalde sequentie $\hat{x}_{k:s-1}$ één zulke kleine stap uit te voeren moeten we eerst voor elke $x_s \in \mathcal{X}_s$ controleren of $\text{mog}_{\hat{x}_{k:s-1} \oplus x_s}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is. Dit kan mits een goede implementatie van het algoritme in constante tijd gebeuren aangezien we in een vorige stap de verzameling $\text{mog}_{\hat{x}_{k:s-1}}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ al bepaald hebben. Vervolgens moeten we voor alle $x_s \in \mathcal{X}_s$

5. Als $s = k$, dan stellen we $\hat{x}_{k:s-1} = \hat{x}_{k:k-1}$ gelijk aan een lege sequentie.

waarvoor $\text{mog}_{\hat{x}_{k:s-1} \oplus x_s}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is controleren of aan de eenvoudige ongelijkheid $\alpha_s^{\max}(x_s) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1} \oplus x_s|z_{k-1})$ voldaan is. We noemen het uitvoeren van deze controles een controlestap en zien dat het aantal bewerkingen voor het uitvoeren van één zulke controlestap lineair is in het aantal mogelijke toestandswaarden. De $x_s \in \mathcal{X}_s$ die na zo'n controlestap aan de beide voorwaarden voldoen noteren we vervolgens als \hat{x}_s en concateneren we met $\hat{x}_{k:s-1}$.

Verderop in deze paragraaf zullen we aantonen dat als we voor een in een vorige stap gevonden sequentie $\hat{x}_{k:s-1}$ de bovenstaande controlestap uitvoeren, we dan steeds minstens één $x_s \in \mathcal{X}_s$ zullen vinden waarvoor $\text{mog}_{\hat{x}_{k:s-1} \oplus x_s}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is én waarvoor $\alpha_s^{\max}(x_s) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1} \oplus x_s|z_{k-1})$. We zullen dus voor elke in een vorige stap gevonden sequentie $\hat{x}_{k:s-1}$ na het uitvoeren van één controlestap gegarandeerd minstens één \hat{x}_s vinden die we met $\hat{x}_{k:s-1}$ kunnen concateneren om de sequentie $\hat{x}_{k:s}$ te vormen. Als we dit kunnen aantonen dan volgt hieruit onmiddellijk dat we door het uitvoeren van een aantal bewerkingen dat lineair is in het aantal toestandswaarden steeds gegarandeerd een \hat{x}_s vinden die we met $\hat{x}_{k:s-1}$ kunnen concateneren.

Het volstaat vervolgens om in te zien dat elke optimale sequentie $\hat{x}_{k:n}$ die we met ons stappenplan bekomen is opgebouwd door voor s gaande van k tot n telkens opnieuw een extra toestand \hat{x}_s toe te voegen aan een tot dan toe al opgebouwde sequentie $\hat{x}_{k:s-1}$. Het toevoegen van een dergelijke extra toestand vergde steeds één controlestap en er werd nooit – zoals vermeld werd in bovenstaande alinea – een controlestap uitgevoerd die geen extra toestand \hat{x}_s opleverde. Hieruit volgt dat het bepalen van één maximale sequentie $\hat{x}_{k:n}$ een aantal controlestappen vergt dat gelijk is aan de lengte van die sequentie. Het bepalen van alle maximale sequenties $\hat{x}_{k:n}$ vergt dus een aantal controlestappen dat lineair is in het aantal maximale sequenties en lineair in de lengte van de sequentie. Eén zulke controlestap uitvoeren is lineair in het aantal toestandswaarden en bijgevolg is het bepalen van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ in het slechtste geval lineair in haar aantal elementen, lineair in de lengte van de beschouwde sequenties en lineair in het aantal mogelijke toestandswaarden.

Het enige wat we nog moeten aantonen is dat als we voor een willekeurige, eerder al opgebouwde sequentie $\hat{x}_{k:s-1}$ een controlestap uitvoeren, we dan ten minste één $x_s \in \mathcal{X}_s$ vinden die aan beide controles voldoet.

Voor $s = k$ volgt dit uit het ongerijmde. Als er geen enkele $x_s \in \mathcal{X}_s$ aan beide controles zou voldoen, dan stopt het stappenplan van paragraaf 6.8.2 en zou $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ geen enkel element bevatten. Dit kan echter niet aangezien er (zie paragraaf 6.3) per definitie altijd minstens één maximaal element moet zijn.

We beschouwen nu een willekeurige $s \in \{k+1, \dots, n\}$.

Aangezien de sequentie $\hat{x}_{k:s-1}$ is opgebouwd met het stappenplan van paragraaf 6.8.2, weten we dat \hat{x}_{s-1} aan de ongelijkheid $\alpha_{s-1}^{\max}(\hat{x}_{s-1}) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1}|z_{k-1})$ voldoet en dat $\text{mog}_{\hat{x}_{k:s-1}}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is.

Uit vergelijking (6.17) volgt dat er zeker één sequentie $x_{s:n}^* \in \mathcal{X}_{s:n}$ bestaat waarvoor $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$. We zullen voor de eerste toestand x_s^* van deze sequentie aantonen dat die aan beide controles voldoet.

Uit de ongelijkheid $\alpha_{s-1}^{\max}(\hat{x}_{s-1}) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1}|z_{k-1})$ kunnen we over de sequentie $x_{s:n}^*$ afleiden dat $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1}|z_{k-1})$, wat via vergelijkingen (6.19) en (6.28) equivalent is met de ongelijkheid $\alpha_s(x_{s:n}^*) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1} \oplus x_s^*|z_{k-1})$. Aangezien door vergelijking (6.19) geldt dat $\alpha_s(x_{s:n}^*) = \alpha_s^{\max}(x_s^*)$, volgt uit het voorgaande dat $\alpha_s^{\max}(x_s^*) \geq \alpha_k^{\text{opt}}(\hat{x}_{k:s-1} \oplus x_s^*|z_{k-1})$ en dat de toestand x_s^* dus aan de eerste controle voldoet.

Om te bewijzen dat x_s^* ook aan de tweede controle voldoet, wat er op neer komt dat $\text{mog}_{\hat{x}_{k:s-1} \oplus x_s^*}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is, volstaat het door definitie (6.16) om aan te tonen dat $\hat{x}_{k:s-1} \oplus x_s^*$ een element is van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$. Hiervoor zullen de onderstaande stellingen een belangrijk hulpmiddel vormen.

Stelling 6.3

Kies een willekeurige $s \in \{1, \dots, n\}$, $\hat{x}_{s-1} \in \mathcal{X}_{s-1}$ en $x_{s:n}^ \in \mathcal{X}_{s:n}$. Als we weten dat $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$, dan volgt hieruit dat $x_{s:n}^* \in \text{opt}(\mathcal{X}_{s:n}|\hat{x}_{s-1}, o_{s:n})$.*

Bewijs

Als $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$, dan weten we door definitie (6.17) dat voor alle $z_{s:n} \in \mathcal{X}_{s:n}$ geldt dat

$$\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) \geq \alpha_{s-1}(\hat{x}_{s-1} \oplus z_{s:n})$$

of dus door vergelijkingen (6.10) en (5.5) dat

$$\bar{S}_{s-1}(\{o_{s-1}\}|\hat{x}_{s-1})\bar{P}_s(\mathbb{I}_{\{x_{s:n}^*\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}) \geq \bar{S}_{s-1}(\{o_{s-1}\}|\hat{x}_{s-1})\bar{P}_s(\mathbb{I}_{\{z_{s:n}\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}).$$

Hieruit volgt op zijn beurt dat voor alle $z_{s:n} \in \mathcal{X}_{s:n}$

$$\bar{P}_s(\mathbb{I}_{\{x_{s:n}^*\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}) \geq \bar{P}_s(\mathbb{I}_{\{z_{s:n}\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}).$$

Aangezien we door coherentie-axioma C3 weten dat

$$\underline{P}_s(-\mathbb{I}_{\{x_{s:n}^*\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}) \geq \underline{P}_s(\mathbb{I}_{\{o_{s:n}\}}(\mathbb{I}_{\{z_{s:n}\}} - \mathbb{I}_{\{x_{s:n}^*\}})|\hat{x}_{s-1}) + \underline{P}_s(-\mathbb{I}_{\{z_{s:n}\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1})$$

en dus door definitie (4.5) dat

$$\underline{P}_s(\mathbb{I}_{\{o_{s:n}\}}(\mathbb{I}_{\{z_{s:n}\}} - \mathbb{I}_{\{x_{s:n}^*\}})|\hat{x}_{s-1}) \leq \bar{P}_s(\mathbb{I}_{\{z_{s:n}\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}) - \bar{P}_s(\mathbb{I}_{\{x_{s:n}^*\}}\mathbb{I}_{\{o_{s:n}\}}|\hat{x}_{s-1}),$$

kunnen we besluiten dat voor alle $z_{s:n} \in \mathcal{X}_{s:n}$ geldt dat

$$\underline{P}_s(\mathbb{I}_{\{o_{s:n}\}}(\mathbb{I}_{\{z_{s:n}\}} - \mathbb{I}_{\{x_{s:n}^*\}})|\hat{x}_{s-1}) \leq 0,$$

wat tot slot door definitie (6.5) impliceert dat $x_{s:n}^* \in \text{opt}(\mathcal{X}_{s:n}|\hat{x}_{s-1}, o_{s:n})$. \square

Stelling 6.4

Beschouw een $k \in \{1, \dots, n-1\}$, $s \in \{k+1, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $\hat{x}_{k:s-1} \in \mathcal{X}_{k:s-1}$. Kies vervolgens een willekeurige $x_{s:n}^* \in \mathcal{X}_{s:n}$ waarvoor $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$. Als we nu weten dat er een $z_{s:n} \in \mathcal{X}_{s:n}$ bestaat waarvoor $\hat{x}_{k:s-1} \oplus z_{s:n}$ een element is van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$, dan volgt hieruit dat ook $\hat{x}_{k:s-1} \oplus x_{s:n}^*$ een element is van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$.

Bewijs

Voor het gemak van notatie stellen we in dit bewijs $\hat{x}_{k-1} = z_{k-1}$.

Uit stelling 6.3 volgt dat $x_{s:n}^* \in \text{opt}(\mathcal{X}_{s:n}|\hat{x}_{s-1}, o_{s:n})$. Hieruit leiden we samen met definitie (6.14) af dat $\hat{x}_{s-1} \oplus x_{s:n}^* \in \text{mog}(\mathcal{X}_{s-1:n}|\hat{x}_{s-2}, o_{s-1:n})$. Als $s = k+1$, dan is hiermee het gestelde al bewezen.

Als $s \in \{k+2, \dots, n\}$, beschouw dan alle $q \in \{k, \dots, s-1\}$ en kijk of er een q bestaat waarvoor – zie definitie (6.15) – $\hat{x}_q \notin \text{Pos}_q(\hat{x}_{q-1})$.

Als er een dergelijke q bestaat, noteer dan de laagste $q \in \{k, \dots, s-2\}$ waarvoor dit het geval is als q^* . Uit definitie (6.14) volgt dan dat zowel $\hat{x}_{q^*:s-1} \oplus x_{s:n}^*$ als $\hat{x}_{q^*:s-1} \oplus z_{s:n}$ een element van $\text{mog}(\mathcal{X}_{q^*:n}|\hat{x}_{q^*-1}, o_{q^*:n})$ zijn omdat alle sequenties in $\hat{x}_{q^*} \oplus \mathcal{X}_{q^*+1:n}$ een element van $\text{mog}(\mathcal{X}_{q^*:n}|\hat{x}_{q^*-1}, o_{q^*:n})$ zijn.

Als er geen $q \in \{k, \dots, s-2\}$ bestaat waarvoor $\hat{x}_q \notin \text{Pos}_q(\hat{x}_{q-1})$, dan stellen we $q^* = s-1$. Er volgt in dat geval uit het herhaaldelijk toepassen van vergelijkingen (6.13) en (6.14) dat $\hat{x}_{s-1} \oplus z_{s:n}$ een element is van $\text{mog}(\mathcal{X}_{s-1:n}|\hat{x}_{s-2}, o_{s-1:n})$ en we vonden in het begin van dit bewijs reeds dat $\hat{x}_{s-1} \oplus x_{s:n}^* \in \text{mog}(\mathcal{X}_{s-1:n}|\hat{x}_{s-2}, o_{s-1:n})$.

We beschikken nu dus over een $q^* \in \{k, \dots, s-1\}$ waarvoor zowel $\hat{x}_{q^*:s-1} \oplus x_{s:n}^*$ als $\hat{x}_{q^*:s-1} \oplus z_{s:n}$ een element van $\text{mog}(\mathcal{X}_{q^*:n}|\hat{x}_{q^*-1}, o_{q^*:n})$ zijn en waarvoor bovendien voor alle $q \in \{k, \dots, q^*-1\}$ geldt dat $\hat{x}_q \in \text{Pos}_q(\hat{x}_{q-1})$. Als $q^* = k$, dan volgt hieruit onmiddellijk het gestelde.

Als $q^* \in \{k+1, \dots, s-1\}$, dan merken we op dat $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ bepaald wordt volgens het recursief principe dat in paragraaf 6.8.1 wordt uiteengezet, namelijk door het herhaaldelijk toepassen van vergelijkingen (6.27) en (6.14). Aangezien we weten dat $\hat{x}_{q^*:s-1} \oplus z_{s:n} \in \text{mog}(\mathcal{X}_{q^*:n}|\hat{x}_{q^*-1}, o_{q^*:n})$ én gegeven is dat $\hat{x}_{k:s-1} \oplus z_{s:n}$ behoort tot $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ volgt hieruit voor alle $q \in \{k, \dots, q^*-1\}$ dat

$$\alpha_q(\hat{x}_{q:s-1} \oplus z_{s:n}) \geq \alpha_q^{\text{opt}}(\hat{x}_q|\hat{x}_{q-1}).$$

Omdat gegeven is dat $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$ weten we door definitie (6.17) dat $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) \geq \alpha_{s-1}(\hat{x}_{s-1} \oplus z_{s:n})$. Hieruit volgt door vergelijking (6.11) voor alle $t \in \{k, \dots, s-1\}$ dat

$$\alpha_t(\hat{x}_{t:s-1} \oplus x_{s:n}^*) \geq \alpha_t(\hat{x}_{t:s-1} \oplus z_{s:n})$$

en dus geldt voor alle $q \in \{k, \dots, q^*-1\}$ dat

$$\alpha_q(\hat{x}_{t:s-1} \oplus x_{s:n}^*) \geq \alpha_q^{\text{opt}}(\hat{x}_q|\hat{x}_{q-1}).$$

Uit dit laatste volgt op zijn beurt – omdat $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ bepaald wordt volgens het recursief principe van paragraaf 6.8.1 en omdat $\hat{x}_{q^*:s-1} \oplus x_{s:n}^*$ een element is van $\text{mog}(\mathcal{X}_{q^*:n}|\hat{x}_{q^*-1}, o_{q^*:n})$ – dat de sequentie $\hat{x}_{k:s-1} \oplus x_{s:n}^*$ behoort tot de verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$, waarmee het gestelde bewezen is. \square

Aangezien $\text{mog}_{\hat{x}_{k:s-1}}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ een niet-lege verzameling is, bestaat er zeker een $z_{s:n} \in \mathcal{X}_{s:n}$ waarvoor $\hat{x}_{k:s-1} \oplus z_{s:n}$ een element is van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$. Bovendien hebben we $x_{s:n}^*$ zodanig gekozen dat $\alpha_{s-1}(\hat{x}_{s-1} \oplus x_{s:n}^*) = \alpha_{s-1}^{\max}(\hat{x}_{s-1})$. Stelling 6.4 leert ons dus dat $\hat{x}_{k:s-1} \oplus x_{s:n}^*$ een element is van $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$.

Als we voor een willekeurige, eerder al opgebouwde sequentie $\hat{x}_{k:s-1}$ een controlestep uitvoeren, zullen we dus wel degelijk op zijn minst één $x_s \in \mathcal{X}_s$ vinden die aan beide controles voldoet en we kunnen bijgevolg – zoals eerder in deze paragraaf reeds verantwoord – besluiten dat het bepalen van $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ met het stappenplan uit paragraaf 6.8.2 inderdaad in het slechtste geval lineair in haar aantal elementen, lineair in de lengte van de beschouwde sequenties en lineair in het aantal mogelijke toestandswaarden is.

6.9 Bespreking van de complexiteit van het algoritme

Het EstiHMM-algoritme bestaat uit twee hoofdonderdelen. In een eerste deel worden alle voorafgaande berekeningen uit paragraaf 6.6 uitgevoerd en in het tweede deel wordt dan met behulp van de in het eerste deel berekende parameters recursief $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ bepaald zoals uiteengezet in paragraaf 6.8.1.

Voor het eerste deel van het algoritme moeten we verschillende parameters berekenen.

Elk van de drempelwaarden $\theta_k(x_k, \hat{x}_k|z_{k-1})$ die door vergelijking (6.23) worden gedefinieerd kan met een beperkt aantal iteraties en dus in constante tijd berekend worden. Voor elke $k \in \{1, \dots, n\}$ en elke combinatie van $x_k \in \mathcal{X}_k$, $\hat{x}_k \in \mathcal{X}_k$ en $z_{k-1} \in \mathcal{X}_{k-1}$ moet een dergelijke drempelwaarde bepaald worden. Het berekenen van al deze waarden is dus kubisch in het aantal mogelijke toestandswaarden en lineair in de lengte van de sequentie.

De parameters $\alpha_k^{\max}(x_k)$ en $\beta_k^{\max}(x_k)$ moeten berekend worden voor elke $k \in \{1, \dots, n\}$ en $x_k \in \mathcal{X}_k$. Verder weten we ook dat het zoeken van het maximum van een lijst met getallen, lineair is in het aantal elementen van die lijst. Doordat we over de recursiebetrekkingen (6.19) en (6.20) beschikken, is het bepalen van alle parameters $\alpha_k^{\max}(x_k)$ en $\beta_k^{\max}(x_k)$ dus kwadratisch in het aantal toestandswaarden en lineair in de lengte van de sequentie.

Tot slot moeten we nog voor elke $k \in \{1, \dots, n\}$, $z_{k-1} \in \mathcal{X}_{k-1}$ en $\hat{x}_k \in \mathcal{X}_k$ de waarde $\alpha_k^{\text{opt}}(\hat{x}_k|z_{k-1})$ bepalen. Ook hier moet telkens het maximum van een lijst gezocht worden en dus is deze berekening lineair in de lengte van de sequentie en kubisch in het aantal mogelijke toestandswaarden.

De complexiteit van het eerste deel van het algoritme is dus kubisch in het aantal mogelijke toestandswaarden en lineair in de lengte van de sequentie.

In het tweede deel van het algoritme bepalen we voor k gaande van n tot 1 telkens voor elke $z_{k-1} \in \mathcal{X}_{k-1}$ de verzameling $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$. We toonden eerder in paragraaf 6.8.3 aan dat het bepalen van zo één verzameling lineair is in de lengte van de beschouwde deelsequentie, lineair in het aantal toestandswaarden en lineair in het aantal elementen – of dus het aantal maximale sequenties – van deze verzameling. We besluiten dat het uitvoeren van dit volledige tweede deel, waarin $\text{opt}(\mathcal{X}_{1:n}|o_{1:n})$ bepaald wordt, kwadratisch is in de lengte van de sequentie, kwadratisch in het aantal mogelijke toestandswaarden, en in essentie⁶ lineair in het aantal bekomen maximale sequenties.

Het uitvoeren van de twee delen heeft dus samen een complexiteit die in het ergste geval kwadratisch is in de lengte van de sequentie, kubisch in het aantal mogelijke toestandswaarden en in essentie lineair in het aantal bekomen maximale toestandsequenties.

Als we de complexiteit van het belangrijkste deel van het EstiHMM-algoritme, namelijk het tweede deel, vergelijken met deze van het Viterbi-algoritme in zijn meest algemene vorm (zie paragraaf 3.4.5), dan zien we dat het verschil in complexiteit relatief beperkt is. Het belangrijkste verschilpunt is dat het Viterbi-algoritme lineair is in de lengte van de sequentie en het EstiHMM-algoritme kwadratisch. De oorzaak hiervan is dat de verschillende oplossingen van het Viterbi-algoritme indifferent zijn (zie paragraaf 3.4.1). Hierdoor hoeven we, in tegenstelling tot bij het EstiHMM-algoritme, voor de elementen van de verzameling $\text{mog}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ niet één voor één te controleren of ze tot $\text{opt}(\mathcal{X}_{k:n}|z_{k-1}, o_{k:n})$ behoren. We kunnen voor het Viterbi-algoritme onmiddellijk vergelijking (3.13) toepassen in plaats van het stappenplan uit te voeren (zie paragraaf 6.8.2) dat we nodig hebben bij het EstiHMM-algoritme. Deze bijkomende complexiteit van het EstiHMM-algoritme lijkt ons echter een redelijke prijs voor de robuustheid die het toelaten van imprecisie ons biedt.

6.10 Implementatie in Python

Het in dit hoofdstuk geïntroduceerde EstiHMM-algoritme werd geïmplementeerd in de programmeertaal Python. Een neerslag van deze code is terug te vinden in bijlage C. De code wordt in hoofdstuk 7 gebruikt om enkele experimenten uit te voeren en vormt tevens de basis voor het uitwerken van de toepassing in hoofdstuk 8.

6. Elke stap, zoals beschreven in paragraaf 6.8.2, heeft een complexiteit die lineair is in het aantal maximale sequenties dat we in dat stadium bekomen.

Enkele experimenten

*“No amount of experimentation can ever prove me right;
a single experiment can prove me wrong.”*

– Albert Einstein

7.1 Inleiding

Uit hoofdstuk 6 blijkt duidelijk dat de verzameling van alle maximale sequenties slechts op efficiënte wijze bepaald kan worden als haar aantal elementen beperkt blijft. Is dit niet het geval, dan zal geen enkel algoritme, hoe goed het ook ontworpen is, in staat zijn om dit probleem te overwinnen.

Omdat het aantal maximale sequenties zo belangrijk is, onderzoeken we dit aantal in dit hoofdstuk van wat dichterbij. We doen dit door te kijken hoe het aantal maximale sequenties varieert in functie van de transitiewaarschijnlijkheden van het model en hoe het evolueert als we de imprecisie laten toenemen. We zullen zien dat het aantal maximale sequenties een interessant gedrag vertoont dat we kunnen verklaren en zelfs tot op zekere hoogte kunnen voorspellen.

7.2 Analyse van het aantal maximale elementen

7.2.1 Beschrijving van een binaire, stationaire iHMM

We beschouwen eerst een binaire, stationaire HMM. Stationariteit werd reeds uitgelegd in paragraaf 2.5 en het binair zijn van een HMM wil gewoon zeggen dat elke toestandsveranderlijke en observatie slechts twee mogelijke toestanden kan aannemen. Laat ons ze aanduiden met 0 en 1.

Het precieze transitie-model kan – zoals in paragraaf 2.6.1 werd aangetoond – volledig bepaald worden door de waarschijnlijkheden om van een bepaalde toestand over te gaan naar een volgende toestand. Men moet hiervoor twee getallen in het eenheidsinterval

kennen: de waarschijnlijkheid p om van toestand 0 naar toestand 0 te gaan, en de waarschijnlijkheid q om van toestand 1 naar toestand 0 te gaan. De waarschijnlijkheid voor een overgang van 0 naar 1 is dan $1 - p$, en die voor een overgang van 1 naar 1 is gelijk aan $1 - q$.

Op analoge wijze kunnen we het marginaal model volledig beschrijven door de waarschijnlijkheid m dat de eerste toestand de waarde 0 aanneemt, omdat de waarschijnlijkheid dat de eerste toestand de waarde 1 aanneemt dan gelijk is aan $1 - m$.

De emissiemodellen kunnen we tot slot op dezelfde manier beschrijven als de transitie-modellen. Ze worden volledig bepaald door de waarschijnlijkheid r om de observatie 0 waar te nemen vanuit een toestand 0 en de waarschijnlijkheid s om de observatie 0 waar te nemen vanuit een toestand 1.

Alle imprecieze modellen worden opgebouwd door contaminatie: we nemen convexe combinaties van de precieze modellen, met combinatiecoëfficiënt $1 - \varepsilon$, en een nietszeggend model, met combinatiecoëfficiënt ε . Een dergelijke combinatie noemen we een lineair-nietszeggende previsie. Een nietszeggend model drukt – zoals de naam al zegt – complete onzekerheid uit en voor een dergelijk model is dus de onder- en bovenprevisie voor een willekeurige functie f respectievelijk gelijk aan $\min f$ en $\max f$. In [2, §2.9.2] wordt aangetoond dat een lineair-nietszeggend model steeds coherent is.

Voor het marginale model vinden we het lineair-nietszeggende model \underline{Q}_1 dat correspondeert met het precieze model Q_1 bijvoorbeeld als volgt: voor elke $f \in \mathcal{L}(\mathcal{X}_1)$ geldt dat

$$\underline{Q}_1(f) = (1 - \varepsilon)Q_1(f) + \varepsilon \min f.$$

Op analoge wijze kan ook het imprecieze transitie-model bepaald worden. In dit experiment houden we het emissiemodel precies en contamineren we het marginale model en het transitie-model met dezelfde combinatiecoëfficiënt ε .¹ Als we ε laten variëren van nul tot één, dan evolueren we van een volledig precieze HMM naar een iHMM met een nietszeggend marginaal en transitie-model en een precies emissiemodel.

7.2.2 De basisprincipes voor een sequentie met lengte twee

We onderzoeken nu het gedrag van een iHMM met lengte twee, waarvan we de onderstaande waarschijnlijkheden vastleggen:²

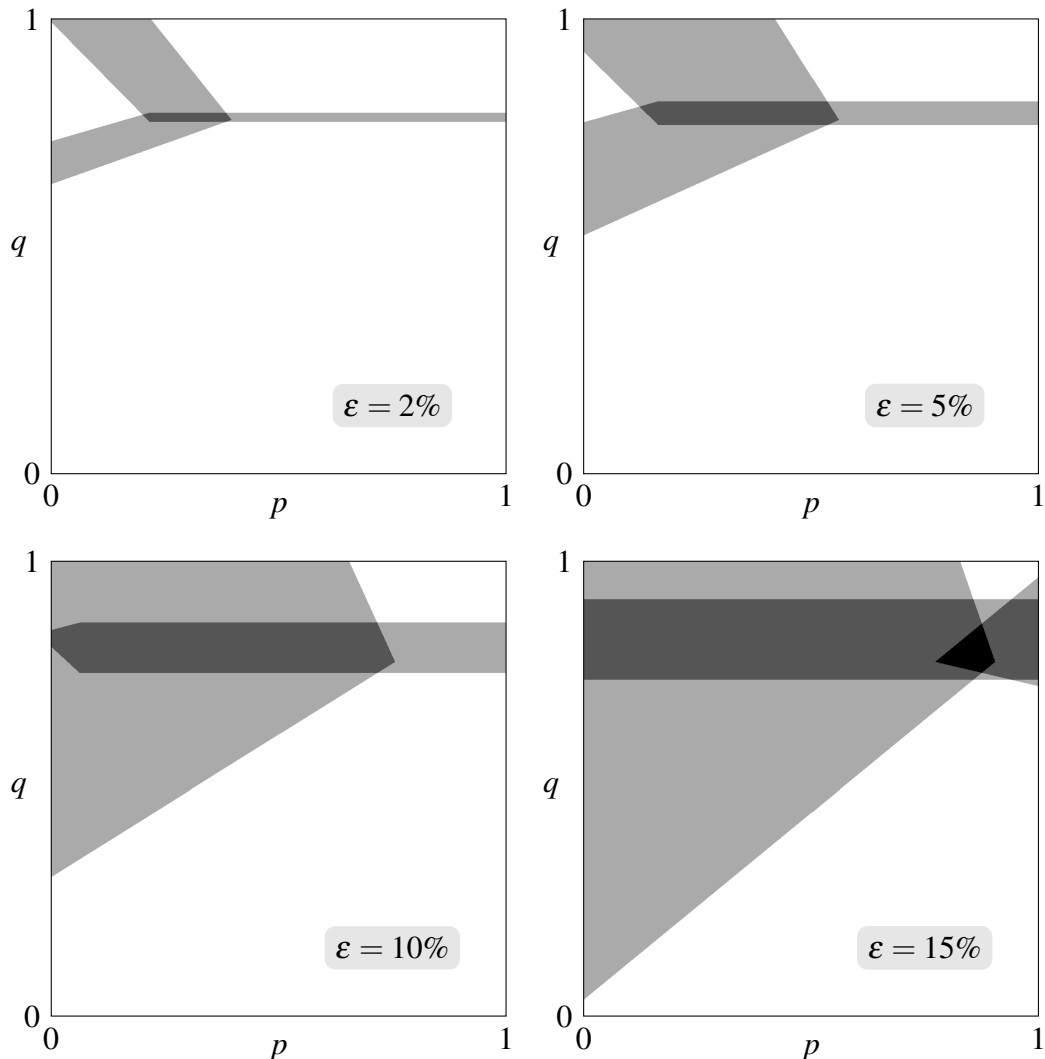
$$m = 0,1, r = 0,8 \text{ en } s = 0,3.$$

Voor een vaste observatiesequentie en een te kiezen waarde voor ε , kunnen we het EstiHMM-algoritme gebruiken om de maximale toestandssequenties te berekenen voor

1. Ook andere mogelijkheden zijn denkbaar. Het zou interessant zijn om deze uit te werken en de resultaten te vergelijken met deze van het hier besproken experiment.

2. Deze keuze is natuurlijk arbitrair. Met andere waarden voor deze waarschijnlijkheden krijgen we andere resultaten, maar de er uit af te leiden conclusie blijft dezelfde.

een willekeurige keuze van p en q binnen het eenheidsinterval. In de onderstaande figuren wordt het aantal maximale sequenties dat we zo vinden voor de observatiesequentie $o_{1:2} = 01$ voorgesteld in functie van p en q . Hoe donkerder het gebied, hoe meer maximale toestandssequenties.



Het aantal maximale toestandssequenties hangt duidelijk sterk af van de transitiewaarschijnslijkheden p en q . In de relatief grote gebieden van de waarschijnlijkheidsruimte die wit zijn gekleurd, treedt slechts één maximale sequentie op – zoals we bij een precieze HMM verwachten –, maar er zijn aangrenzende gebieden waar we een groter aantal maximale sequenties krijgen. Lichtgrijze gebieden corresponderen met twee maximale sequenties en donkergrijze met drie. In dit voorbeeld (een binaire keten met lengte twee) is het hoogst mogelijke aantal maximale sequenties natuurlijk vier. Gebieden waar dit aantal bereikt wordt zijn zwart ingekleurd.

Uit de figuren blijkt duidelijk wat er gebeurt als we ϵ – de imprecisie dus – laten toenemen: de grijze gebieden breiden uit en voor vaste waarden van p en q neemt het aantal maximale sequenties toe. Voor $\epsilon = 15\%$ vinden we zelfs een klein gebied waar elk

van de vier mogelijke toestandssequenties maximaal is: in dat gebied kunnen we door de relatief grote imprecisie van de lokale modellen enkel besluiten dat we over onvoldoende informatie beschikken om een inschatting te maken van de toestandssequentie die de waargenomen observatiessequentie $o_{1:2} = 01$ heeft gegenereerd.

Voor kleine waarden van ε zien we dat de gebieden met meer dan één maximale sequentie behoorlijk klein zijn en dat ze lijken op stroken die tot lijnen versmallen naarmate ε tot nul nadert, of naarmate het model dus preciezer wordt. Dit suggereert dat we op zijn minst kwalitatief zouden moeten kunnen verklaren en/of voorspellen waar deze zones met grotere imprecisie vandaan komen, door te kijken naar de precieze modellen waarop onze imprecieze modellen zijn gebaseerd. De zones waar een iHMM verschillende (onderling onvergelijkbare) maximale toestandssequenties oplevert, lijken namelijk verbrede versies van de lijnen van indifferentie die optreden bij de corresponderende HMM.

Met een lijn van indifferentie bedoelen we de verzameling (of meetkundige plaats) van transitiewaarschijnelijkheden (p, q) waarvoor twee gegeven toestandssequenties $x_{1:2}$ en $\hat{x}_{1:2}$ een zelfde waarschijnlijkheid hebben, conditioneel op de waargenomen observaties:

$$p(x_{1:2}|o_{1:2}) = p(\hat{x}_{1:2}|o_{1:2}).$$

Deze gelijkheid kunnen we, als $p(o_{1:2}) > 0$, met vergelijking (3.4) omvormen tot de equivalente uitdrukking

$$p(x_{1:2}, o_{1:2}) = p(\hat{x}_{1:2}, o_{1:2}).$$

$p(o_{1:2})$ is in ons voorbeeld steeds groter dan nul, omdat alle emissiewaarschijnelijkheden verschillend zijn van nul, en dus kunnen we de bovenstaande gelijkheid gebruiken om de lijnen van indifferentie te bepalen.

Voor onze observatiessequentie $o_{1:2} = 01$ vinden we de onderstaande uitdrukkingen voor $p(z_{1:2}, o_{1:2})$, waarbij we $z_{1:2}$ achtereenvolgens gelijk stellen aan één van de vier mogelijke toestandssequenties:

$$\begin{aligned} p(00, 01) &= mr(1-r)p \\ p(01, 01) &= mr(1-s)(1-p) \\ p(10, 01) &= (1-m)s(1-r)q \\ p(11, 01) &= (1-m)s(1-s)(1-q). \end{aligned}$$

Door telkens twee van de bovenstaande uitdrukkingen gelijk te stellen, drukken we uit dat de corresponderende twee toestandssequenties een gelijke waarschijnlijkheid hebben, conditioneel op de observaties. Voor elk van de zes mogelijke combinaties bekommen we zo een vergelijking die een functie is van p en q en een lijn van indifferentie definieert. Deze lijnen zijn, samen met de toestandssequenties waarmee ze corresponderen, weergegeven op de onderstaande figuur.

Delen van deze indifferentielijnen, in het vet aangeduid, delen de waarschijnlijkheidsruimte op in drie zones, waarin de toestandssequenties 01, 10 en 11 maximaal zijn (ze hebben de hoogste waarschijnlijkheid, conditioneel op de waargenomen observaties). De sequentie 00 blijkt in dit voorbeeld nooit maximaal te zijn.

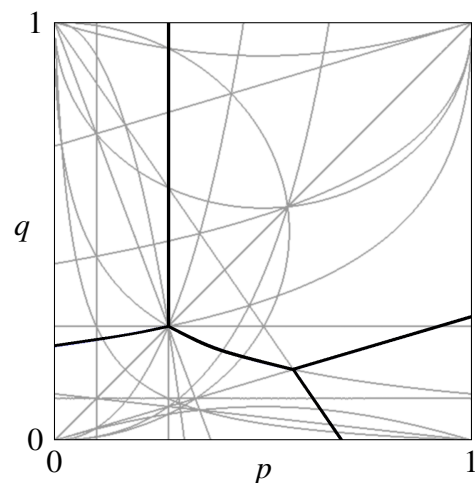
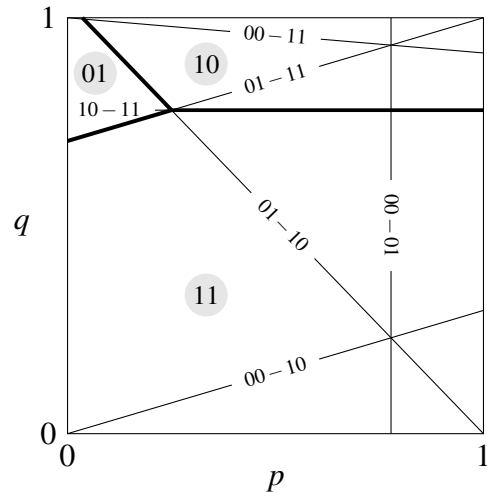
Op de vette lijnen zijn telkens twee van deze sequenties maximaal en ter hoogte van hun snijpunt zijn ze alle drie maximaal. Merk op dat dit mooi illustreert wat we in paragraaf 3.4 vermeldden, namelijk dat er – in tegenstelling tot waar men vaak impliciet van uit gaat – wel degelijk meerdere maximale toestandssequenties kunnen zijn, ook in het precieze geval.

De gelijkensissen tussen de aan de hand van enkele eenvoudige vergelijkingen gevonden lijnen van indifferentie voor een precies model en de zones met meerdere maximale elementen die we vonden voor een imprecies model is treffend. Dit kan een goede kwalitatieve manier zijn om te voorspellen of een precieze gevolgtrekking al dan niet robuust zal zijn.

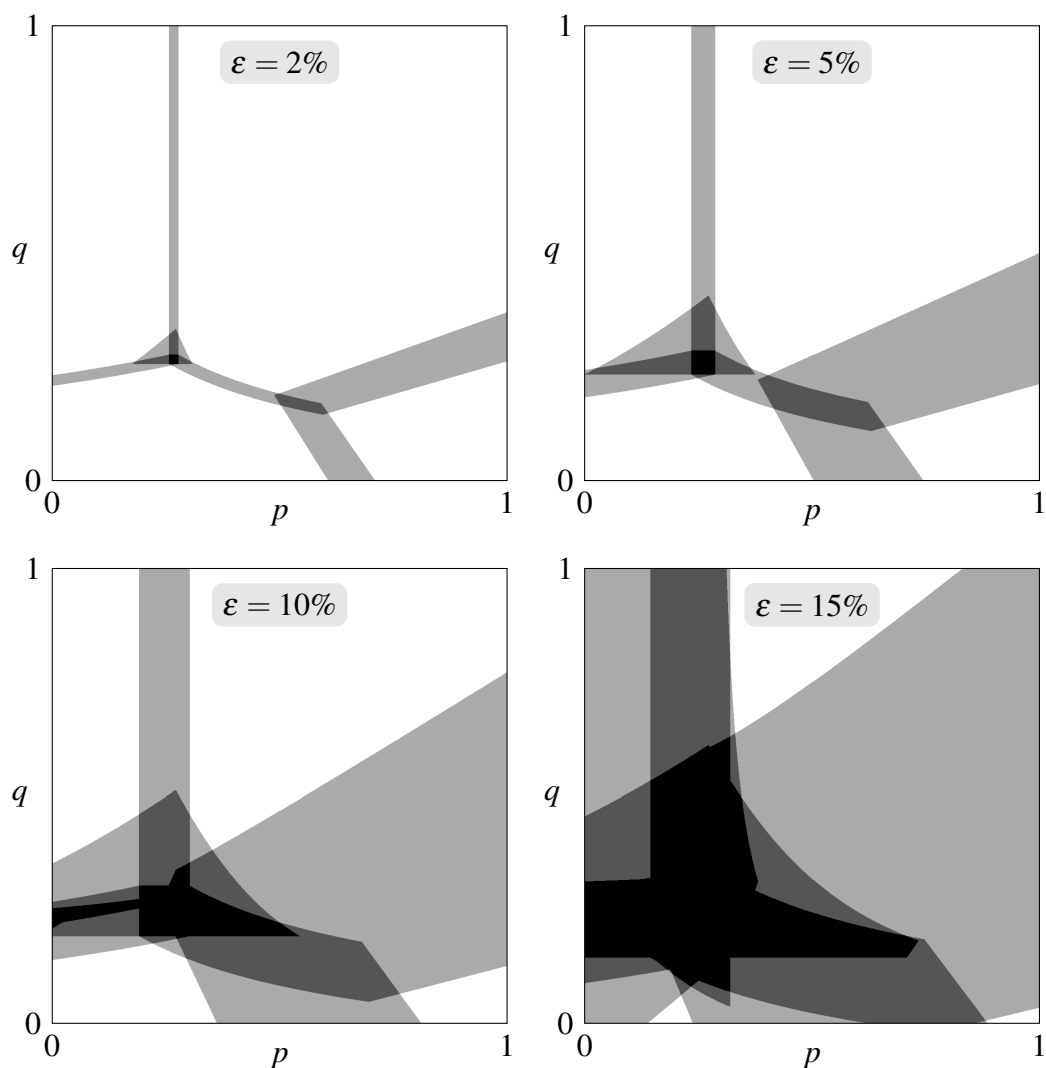
De hier bekomen resultaten zijn bovendien een bevestiging van de correcte werking van het EstiHMM-algoritme, aangezien we naarmate de imprecisie van het model afneemt – zoals het hoort – convergeren naar de resultaten die we met een precies model zouden vinden volgens de reeds bestaande algoritmes voor precieze modellen, zoals het Viterbi-algoritme.

7.2.3 Resultaten voor een sequentie met lengte drie

Voor een keten met lengte drie kunnen we op een volledig analoge manier de indifferentielijnen bepalen die corresponderen met een precies model. Als we hetzelfde marginaal model en emissiemodel gebruiken als in het vorige voorbeeld, dan zien de indifferentielijnen voor de observatiesequentie 000 er uit zoals op de nevenstaande figuur. De vette delen van de indifferentielijnen delen de waarschijnlijkheidsruimte opnieuw op in zones waar verschillende toestandssequenties maximaal zijn en zijn zelf de plaats waar twee of meer toestandssequenties maximaal zijn.



Als we dit vergelijken met de onderstaande visualisaties van het aantal maximale toestandssequenties voor een corresponderend imprecies model, dan is de gelijkens opnieuw duidelijk zichtbaar. De zwarte zones in de figuren zijn ditmaal gebieden met ten minste vier optimale sequenties, aangezien het maximale aantal (voor deze binaire keten met lengte drie) acht bedraagt.



7.3 Een illustratie van de efficiëntie van het algoritme

Om te demonstreren dat het EstiHMM-algoritme wel degelijk efficiënt is, laten we het de maximale sequenties bepalen voor een willekeurige observatiesequentie met lengte 100.

We beschouwen opnieuw dezelfde binaire, stationaire HMM als eerder in dit hoofdstuk, maar ditmaal met de volgende precieze marginale en emissiewaarschijnlijkheden:

$$m = 0,1, r = 0,98 \text{ en } s = 0,01.$$

Een toepassing

“Het is niet wat het is, 't is wat je der mee doet.”

– Bart Peeters in *De Hemel in het Klad*

8.1 Inleiding

Op café¹ krijg ik naast klassiekers zoals “Wat studeer je?”, “Wat drinkt ge van me?” en “Weet gij het toilet hier zijn?” soms ook de vraag “Waar gaat je thesis over?”. Wie dat enkel uit beleefdheid vraagt en eigenlijk niet geïnteresseerd is, kan ik bij deze beter even waarschuwen: u komt bedrogen uit. Enthousiast begin ik dan namelijk uitgebreid te vertellen over het modelleren van de werkelijkheid, waarom het belangrijk is om imprecisie toe te laten en hoe die dingen samen komen in een imprecies hidden Markovmodel. Wie mij kent kan het zich levendig voorstellen, of heeft het ongetwijfeld al meegemaakt. Na enkele minuten stelt dan meestal iemand de vraag: “Wat kunt ge daar nu eigenlijk mee doen?”. Voor die mensen is dit hoofdstuk geschreven.

8.2 Omschrijving van het probleem

Een goed boek wil ik vastnemen. Ik wil de geur van vergeelde, licht krakende bladzijden opsnuiven, me vervolgens in een schommelstoel nestelen en de rest van de wereld langzaam laten vervagen . . . Wie weet herkent u dit gevoel en is de charme van een tastbare verzameling boeken u niet onbekend. U zult het echter met mij eens zijn dat u die dingen niet onbeperkt kan verzamelen: opslagruimte is – zoals vele dingen² – eindig. Daarom worden in deze moderne tijden steeds meer teksten gedigitaliseerd. Hoewel men ze simpelweg zou kunnen inscannen en zo opslaan op een PC, wil men

1. Al kan het evengoed een willekeurige andere sociale gelegenheid zijn, ik wil hier geen verkeerd beeld van mezelf schetsen.

2. Einstein zei ooit dat slechts twee dingen oneindig zijn: het universum en de menselijke stomheid. Over het eerste was hij niet zeker. . .

vaak meer dan dat. Men wil de tekst bijvoorbeeld ook kunnen bewerken of er efficiënt in zoeken. Daarvoor is meer nodig dan het inscannen van de documenten. Men moet de afzonderlijke letters van de tekst herkennen en zo een tekstbestand creëren. Dit heeft niet alleen als voordeel dat men de tekst dan kan bewerken, maar ook dat dit bestand veel minder geheugen inneemt dan een ingescand document.

Het afzonderlijk herkennen van de verschillende letters van een tekst (men spreekt van karakterherkenning) gebeurt door een OCR-programma (Optical Character Recognition). Die programma's werken zeer goed en worden wereldwijd met succes toegepast. Toch komt het soms voor dat een OCR-programma fouten maakt. Vergissen is, zo blijkt, niet enkel menselijk. Een boek dat oorspronkelijk over VOGELS ging, kan na verwerking door een OCR-programma plots een boeiend meesterwerk over VOGEIS zijn.

Dit willen we natuurlijk vermijden, en er zijn hier allerlei oplossingen voor. Je kan het hele boek zelf lezen, vergelijken met de uitkomst van het OCR-programma en manueel de fouten corrigeren, maar dit lijkt mij enkel besteed aan mensen zonder hobby's.³ Je kan een computer ook laten kijken of de ingelezen woorden voorkomen in het woordenboek. Op die manier zou je waarschijnlijk merken dat VOGEIS geen bestaand woord is, maar je weet dan nog niet welk woord het wel moet zijn. Jij, als mens, beseft waarschijnlijk snel dat het eigenlijk VOGELS moet zijn, maar dat is voor een computer niet zo vanzelfsprekend. Voor dit ene woord kan je natuurlijk zelf snel eens kijken en VOGEIS in VOGELS veranderen, maar als je een hele bibliotheek hebt gedigitaliseerd, dan zou de lijst met woorden die niet in het woordenboek voorkomen wel eens te lang kunnen worden om te laten verbeteren door mensen met een normale vervelingsdrempel. Bovendien zijn er heel wat woorden die niet in het woordenboek voorkomen, maar die toch correct zijn. Komt MARKOVMODEL in uw woordenboek voor?

In het vervolg van dit hoofdstuk zullen we een andere manier bespreken om dit probleem op te lossen. Met behulp van een al dan niet imprecies hidden Markovmodel en de in deze masterproef besproken algoritmes zijn we namelijk in staat om fout ingelezen woorden te corrigeren. Ik was zelf oprecht verbaasd toen ik de resultaten in dit hoofdstuk voor de eerste keer zag, en hoop dat u dezelfde verwondering te wachten staat.

8.3 De gebruikte data

We zullen de oplossing van ons probleem illustreren aan de hand van Dante's meesterwerk *La Divina Commedia*. We gebruiken het tweede canto van het boekdeel Inferno (de *modeltekst*) om een hidden Markovmodel op te bouwen. Vervolgens gebruiken we dit model om de eerste negen terzines van het eerste canto (de *voorbeeldtekst*), nadat ze door een OCR-programma zijn ingelezen, te proberen verbeteren.

3. Of mensen waarvan net dít de hobby is.

De volledige tekst van het tweede canto is terug te vinden in bijlage B.1. Hoewel OCR-programma's in werkelijkheid ook leestekens, accenten en andere tekens kunnen inlezen, zullen we ons voor dit illustratief voorbeeld beperken tot de letters van het alfabet. We behandelen de tekst bovendien woord per woord en vormen daarom voor de rest van de analyse de originele tekst om tot een genormaliseerde versie (zie bijlage B.2), die enkel letters van het alfabet bevat. We maken ook geen onderscheid tussen hoofdletters en kleine letters, en zetten daarom alle kleine letters om in hoofdletters.

Onze genormaliseerde modeltekst wordt vervolgens verwerkt door een OCR-programma, waarna we opnieuw dezelfde tekst krijgen, waar nu enkele fouten zijn ingeslopen (zie bijlage B.3). Een lijst met de originele woorden en de corresponderende eventueel foutief ingelezen woorden die het OCR-programma geeft, is terug te vinden in bijlage B.4. Omdat het nogal bewerkelijk is om bij herhaalde experimenten de beschouwde tekst telkens af te drukken, in te scannen en vervolgens door een OCR-programma te laten verwerken, hebben we de werking van een OCR-programma gesimuleerd met een virtueel OCR-programma. Dit heeft bovendien als voordeel dat we het OCR-programma op die manier veel meer fouten kunnen laten maken dan in werkelijkheid, zodat we de werking van het algoritme beter kunnen illustreren.

In ons virtuele OCR-programma heeft elke letter een even grote kans om foutief ingelezen te worden. Voor het in dit hoofdstuk uitgewerkte voorbeeld heeft elke letter hiervoor een kans van 1 op 10. Als een letter foutief ingelezen wordt, dan geeft het virtuele OCR-programma volkomen lukraak een andere letter uit een lijst met mogelijke foutieve letters die lijken op de originele letter. De letter P zal bijvoorbeeld als hij foutief ingelezen wordt vervangen worden door een B, een F of een R. De mogelijke foutieve letters voor alle letters uit het alfabet zijn terug te vinden in de onderstaande tabel.

A	H, N, R	H	A, B, N, R	Q	D, G, O, U
B	E, H, P, R	I	L, T, V, Z	R	A, B, E, H, P
C	D, O, U	L	I, Z	S	E, G
D	C, O, Q	M	A, N	T	F, I
E	B, F, R, S	N	H, M	U	C, O, Q, V
F	E, P, T	O	C, D, G, Q, U	V	I, U
G	O, Q, S	P	B, F, R	Z	I, L

Nadat we aan de hand van het tweede canto een model hebben opgebouwd, zullen we dit gebruiken om de eerste negen terzines van het eerste canto te verbeteren, nadat ze zijn ingelezen door hetzelfde virtuele OCR-programma. De volledige tekst van die eerste negen terzines is terug te vinden in bijlage B.7. De genormaliseerde versie hiervan staat in bijlage B.8. Nadat deze genormaliseerde versie door een OCR-programma is verwerkt krijgen we de foutieve tekst (zie bijlage B.9) die we zullen proberen te verbeteren. Een lijst met de originele woorden van de voorbeeldtekst en de corresponderende eventueel foutief ingelezen woorden die het OCR-programma geeft vind je in bijlage B.10.

8.4 Observaties en verborgen toestanden

Een eerste stap in het modelleren van onze situatie, is het identificeren van de verborgen toestanden en de observaties.

De verborgen toestandssequentie is in ons voorbeeld het woord zoals het in de Divina Commedia staat. Dit woord is een sequentie van letters, die hier de mogelijke toestanden zijn. De toestandsverzameling \mathcal{X} is dus het alfabet:

$$\mathcal{X} = \{A, B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R, S, T, U, V, Z\}.$$

Merk op dat dit alfabet slechts 21 letters telt omdat in het Italiaans de letters J, K, W, X en Y niet voorkomen. Aangezien we de tekst vereenvoudigd hebben tot een woordenlijst die geen speciale symbolen zoals accenten of leestekens bevat, hoeven we deze ook niet te beschouwen in onze toestandsverzameling. We maken ook geen onderscheid tussen kleine letters en hoofdletters. In een realistische toepassing zouden we wel met al deze dingen rekening moeten houden, maar aangezien dit slechts een illustratief voorbeeld is, kiezen we er bewust voor om dit niet te doen, omdat het ons toch maar zou afleiden van de essentie van het probleem.

De geobserveerde sequentie is in dit probleem hetzelfde woord, maar nadat het door het OCR-programma is ingelezen, waardoor er dus eventueel enkele fouten zijn ingeslopen. De verzameling van mogelijk observaties \mathcal{O} is in dit geval dus gelijk aan de toestandsverzameling \mathcal{X} . Hoewel deze beide verzamelingen vaak verschillen, zoals het geval was bij het voorbeeld in hoofdstuk 3 over het weer buiten en de deur die klemt, blijkt dit dus zeker niet altijd het geval te hoeven zijn.

Hoewel dit intuïtief duidelijk is, vermelden we voor de volledigheid ook nog dat in dit voorbeeld, voor alle $k \in \{1, \dots, n\}$, $\mathcal{X}_k = \mathcal{X}$ en $\mathcal{O}_k = \mathcal{O}$.

8.5 Modelling met een HMM

8.5.1 Lokale precieze modellen opstellen

De lokale modellen waaruit een precies hidden Markovmodel opgebouwd is, worden – zoals in paragraaf 2.6.1 werd aangetoond – volledig bepaald door de waarschijnlijkheid van de verschillende mogelijke toestandswaarden. Deze waarschijnlijkheden stellen we gelijk aan de relatieve frequentie waarmee de beschouwde toestandswaarde voorkomt in de woordenlijst (bijlage B.4) waarmee we ons model opbouwen.

Het marginale model is een lokaal model voor de eerste letter van een woord. De hiervoor vereiste waarschijnlijkheden bekomen we door in de lijst met originele (vóór de verwerking door het OCR-programma) woorden te tellen hoe vaak elke letter in het alfabet voorkomt als eerste letter van een woord. De resultaten van deze telling zijn terug te vinden in bijlage B.5.1. Het aantal keer dat een toestand $x \in \mathcal{X}$, met

bijvoorbeeld $x = A$, optreedt als beginletter noteren we als n_x . Zo blijkt uit bijlage B.5.1 bijvoorbeeld dat $n_A = 67$. De waarschijnlijkheid $Q_1(\{A\}) = q_1(A)$ dat A de beginletter is van een willekeurig woord is dan

$$Q_1(\{A\}) = \frac{n_A}{\sum_{z \in \mathcal{X}} n_z} = \frac{67}{1018} = 6,58\% = 0,0658.$$

De op die manier gevonden waarschijnlijkheid is voor elke mogelijke beginletter terug te vinden in bijlage B.6.1.

Het transitie-model wordt op een gelijkaardige manier opgebouwd. We nemen aan dat het transitie-model onafhankelijk is van de positie in de sequentie (of in dit geval dus het woord) en dat we dus te maken hebben met een stationair model (zie paragraaf 2.5). Het aantal keer dat een toestand $y \in \mathcal{X}$ in een sequentie onmiddellijk gevolgd wordt door een toestand $x \in \mathcal{X}$ noteren we als $n_{x|y}$. In bijlage B.5.2 zien we bijvoorbeeld dat $n_{T|A} = 22$. In de gehele woordenlijst waarmee we ons model opbouwen kwam het dus 22 keer voor dat een letter A in een woord onmiddellijk gevolgd werd door een letter T. De waarschijnlijkheid $Q(\{T\}|A) = q(T|A)$ dat de letter T voorkomt in een woord, conditioneel op het feit dat de voorgaande letter een A was, wordt dan gegeven door

$$Q(\{T\}|A) = \frac{n_{T|A}}{\sum_{z \in \mathcal{X}} n_{z|A}} = \frac{22}{240} = 9,17\% = 0,0917.$$

Het resultaat van deze berekening is, voor elke mogelijke combinatie van twee opeenvolgende letters, terug te vinden in bijlage B.6.2.

Het emissie-model wordt tot slot verkregen door elke letter in de oorspronkelijke woordenlijst te vergelijken met de corresponderende letter in de door het OCR-programma ingelezen woordenlijst. Het aantal keer dat een toestand $x \in \mathcal{X}$ aanleiding geeft tot een observatie $o \in \mathcal{O}$ noteren we als $m_{o|x}$. Zo zien we in bijlage B.5.3 bijvoorbeeld dat $m_{Z|L} = 12$, of dus dat de letter L in totaal 12 keer foutief werd ingelezen als een Z door het OCR-programma. De waarschijnlijkheid $S(\{Z\}|L) = q(Z|L)$ dat de letter Z wordt geobserveerd, conditioneel op het feit dat de oorspronkelijke verborgen toestand een L was, wordt gegeven door

$$S(\{Z\}|L) = \frac{m_{Z|L}}{\sum_{z \in \mathcal{O}} m_{z|L}} = \frac{12}{253} = 4,74\% = 0,0474.$$

Het resultaat van deze berekening is, voor elke mogelijke combinatie van een toestand en een observatie, terug te vinden in bijlage B.6.3.

8.5.2 Resultaten van het Viterbi-algoritme

Met de in de vorige paragraaf opgebouwde lokale precieze modellen, kunnen we nu voor elk woord dat door het OCR-programma is verwerkt, en dus eventueel fouten bevat, een poging doen om deze fouten via het in paragraaf 3.4 besproken Viterbi-algoritme te verbeteren. We doen dit door het eventueel foutieve woord als een geobserveerde

sequentie te beschouwen en het onderliggende – niet gekend veronderstelde – originele woord als verborgen toestandssequentie te interpreteren. We bepalen vervolgens via het Viterbi-algoritme de optimale sequenties die overeenstemmen met het geobserveerde woord en beschouwen deze als een schatting van het werkelijke originele woord. Merk op dat het originele correcte woord in deze berekening zelf nooit gebruikt wordt. In realiteit kent men het originele woord zelfs niet. Het is net de bedoeling om dit originele woord te achterhalen via het Viterbi-algoritme, uitgaande van het eventueel foutieve woord dat door het OCR-programma wordt geleverd. We gebruiken de originele woorden hier enkel ter controle achteraf, om te kijken in hoeverre we in ons opzet geslaagd zijn.

In bijlage B.11 is voor elk woord van de te verbeteren tekst het resultaat van deze berekening terug te vinden. In de eerste kolom vind je steeds het originele correcte woord terug. In de tweede kolom staat ditzelfde woord, maar nadat het is ingelezen door het OCR-programma, waardoor het in sommige gevallen fouten bevat. De derde kolom geeft tot slot de woorden die door het EstiHMM-algoritme gesuggereerd worden (de maximale toestandssequenties) ter verbetering van het woord dat door het OCR-programma is ingelezen (de observatiesequentie). De suggesties (de maximale sequenties) van het Viterbi-algoritme zijn – zoals in paragraaf 6.3 werd aangetoond – steeds vervat in de suggesties van het EstiHMM-algoritme en we markeren deze oplossingen van het Viterbi-algoritme door ze te onderlijnen.

Een eerste opmerking die we bij de resultaten in bijlage B.11 geven, is dat het Viterbi-algoritme voor de door ons verbeterde tekst steeds slechts één optimale oplossing suggereert per woord. Dit hoeft niet te verbazen, aangezien het – zoals in 3.4.2 al werd vermeld – eerder uitzonderlijk is om in het precieze geval meerdere optimale sequenties te bekomen. De resultaten van het EstiHMM-algoritme worden verderop in dit hoofdstuk besproken. In deze paragraaf beperken we ons tot een bespreking van de door het Viterbi-algoritme gesuggereerde oplossingen.

Om de resultaten makkelijker te kunnen interpreteren, hebben we in bijlage B.11 elk woord dat overeenstemt met het oorspronkelijke correcte woord in het vet aangeduid. Zo kunnen we bijvoorbeeld makkelijk zien dat een woord correct is ingelezen door het OCR-programma, doordat het in de tweede kolom in het vet is aangeduid. Als het Viterbi-algoritme een foutieve oplossing voorstelt, dan kan je dit snel opmerken omdat het onderlijnde woord in de derde kolom niet in het vet is aangeduid. We geven een voorbeeld ter verduidelijking:

QUANTO OUANTO CUANTO DUANTO FUANTO **QUANTO**

Het woord QUANTO werd hier door het OCR-programma foutief ingelezen. In plaats van QUANTO, staat er in de gedigitaliseerde versie van de tekst OUANTO. Als we dit als observatiesequentie opgeven aan het precies model dat we hebben opgebouwd, dan suggereert het Viterbi-algoritme het woord DUANTO als optimale sequentie. Het Viterbi-algoritme slaagt er in dit geval dus niet in om de gemaakte fout opnieuw

te corrigeren. Het EstiHMM-algoritme slaagt er wel in om de correcte oplossing te suggereren, maar die resultaten zullen we pas verderop in dit hoofdstuk bespreken. We beperken ons voorlopig tot de oplossing van het Viterbi-algoritme en geven in het vervolg van deze paragraaf de oplossingen van het EstiHMM-algoritme niet weer. Bovenstaand voorbeeld wordt dus als volgt genoteerd:

QUANTO OUANTO DUANTO

Ook voor de andere voorbeelden in deze paragraaf, worden de oplossingen van het EstiHMM-algoritme niet afgebeeld. Desgewenst kan men ze terugvinden in bijlage B.11.

De resultaten van het Viterbi-algoritme worden samengevat in de onderstaande tabel:

Viterbi	TOTAAL	juist na OCR	fout na OCR
TOTAAL	200 (100%)	137 (68,5%)	63 (31,5%)
juist na Viterbi	157 (78,5%)	132	25
fout na Viterbi	43 (21,5%)	5	38

De belangrijkste conclusie die we uit deze tabel kunnen trekken is dat de tekst nadat elk woord is vervangen door de suggestie van het Viterbi-algoritme minder fouten bevat dan voorheen. De te verbeteren tekst, zoals hij door het OCR-programma wordt gegeven, bevat 68,5% correcte woorden. Nadat we het Viterbi-algoritme hier op losgelaten hebben is de hoeveelheid correcte woorden gestegen tot 78,5%.

De tabel onderscheidt verder vier verschillende categorieën. We geven voor elke van deze categorieën enkele voorbeelden ter illustratie.

Van de in totaal 200 woorden die we beschouwen, werden er 63 foutief ingelezen door het OCR-programma. Het zijn deze woorden die we met het Viterbi-algoritme willen proberen te detecteren en verbeteren. Van deze 63 woorden waren er 25 waarvoor het Viterbi-algoritme er in slaagde om het oorspronkelijke, correcte woord te suggereren als beter alternatief dan het door het OCR-programma gegeven woord. Enkele geslaagde correcties zijn bijvoorbeeld:

CAMMIN	CHMMIN	<u>CAMMIN</u>
CHE	OHS	<u>CHE</u>
LA	LR	<u>LA</u>
TERMINAVA	TERMLNAVA	<u>TERMINAVA</u>
VOLSE	VOLSB	<u>VOLSE</u>

We zien dat het Viterbi-algoritme, aan de hand van de informatie die in de lokale precieze modellen vervat zit, er in slaagt om taalkundig onlogische letteroepvolgingen te detecteren en deze vervolgens te vervangen door een “logischer” opeenvolging. Op die manier kunnen we een groot aantal van de door het OCR-programma gegenereerde

fouten opnieuw corrigeren.

Van de 63 woorden die foutief zijn ingelezen door het OCR-programma, waren er 38 waarvoor het Viterbi-algoritme er niet in slaagde om deze fouten te corrigeren. In sommige gevallen stelde het Viterbi-algoritme voor om geen correctie uit te voeren:

FORTE	PORTE	<u>PORTE</u>
TANTO	IANIO	<u>IANIO</u>
DRITTO	ORITIO	<u>ORITIO</u>
RETRO	BEIRO	<u>BEIRO</u>

In andere gevallen werd wel een correctie voorgesteld, maar leverde deze niet het correcte originele woord op:

TRATTAR	TAATTAR	<u>TARTTAR</u>
IO	ZO	<u>LO</u>
PUNTO	PUHTO	<u>PUATO</u>
ABBANDONAI	ABBANDONAZ	<u>ABBANDONAL</u>

Van de 200 behandelde woorden, werden er 137 correct ingelezen door het OCR-programma. Hier is dus geen verbetering nodig. Van deze 137 woorden waren er 132 waarvoor het Viterbi-algoritme dit bevestigde en het originele, correcte woord suggereerde. We geven enkele voorbeelden:

NEL	NEL	<u>NEL</u>
VITA	VITA	<u>VITA</u>
SMARRITA	SMARRITA	<u>SMARRITA</u>

Van de 137 correct ingelezen woorden waren er echter ook 5 waarvan het Viterbi-algoritme – onterecht – dacht dat het woord verbeterd moest worden en een foutieve oplossing suggereerde:

MEZZO	MEZZO	<u>MEZIO</u>
EH	EH	<u>EN</u>
HO	HO	<u>NO</u>
AFFANNATA	AFFANNATA	<u>ATTANNATA</u>
ACQUA	ACQUA	<u>ACOVA</u>

Dit soort resultaten is natuurlijk zeer ongewenst, aangezien we op deze manier extra fouten introduceren in plaats van foutieve woorden te verbeteren. We zullen verderop in dit hoofdstuk aantonen dat het EstiHMM-algoritme dit probleem in belangrijke mate kan verhelpen.

8.6 Modelling met een iHMM

8.6.1 Lokale imprecieze modellen opstellen

Om de lokale modellen waaruit een iHMM bestaat te bepalen, maken we gebruik van het imprecies Dirichletmodel (IDM) [19, §2.3]. Dit model geeft ons in functie van de in paragraaf 8.5.1 vermelde tellingen n_x , $n_{x|y}$ en $m_{o|x}$ een uitdrukking voor de lokale boven- en onderwaarschijnlijkheden van de verschillende mogelijke toestandswaarden.

Voor het marginale model geeft het IDM ons de onderstaande twee uitdrukkingen, die geldig zijn voor elke $x \in \mathcal{X}$:

$$\underline{Q}_1(\{x\}) = \frac{n_x}{\sum_{z \in \mathcal{X}} n_z + s} \text{ en } \bar{Q}_1(\{x\}) = \frac{n_x + s}{\sum_{z \in \mathcal{X}} n_z + s},$$

waarbij s een parameter van het IDM is die een maat is voor de imprecisie van de modellen en dus ook voor de graad van robuustheid die in de daarmee gemaakte gevolgtrekkingen ingebouwd wordt. Deze parameter kan in principe volkomen vrij gekozen worden. We kiezen er in onze toepassing voor om s gelijk te stellen aan twee. Voor de toestand $x = A$ vinden we zo dat $\underline{Q}_1(\{A\}) = 6,57\%$ en $\bar{Q}_1(\{A\}) = 6,76\%$. Zoals te verwachten viel, is de precieze waarschijnlijkheid $Q_1(\{A\}) = 6,58\%$ gelegen tussen de corresponderende onder- en bovenwaarschijnlijkheden van het imprecieze model. De resultaten van deze berekeningen zijn voor elke $x \in \mathcal{X}$ terug te vinden in bijlage B.6.1.

Een mooie eigenschap van het IDM is dat de imprecisie van de modellen, als men de parameter s eenmaal gekozen heeft, toeneemt naarmate er minder data beschikbaar zijn om het model op te bouwen. Dit is een intuïtief aantrekkelijke eigenschap, die jammer genoeg niet aanwezig is bij precieze waarschijnlijkheidsmodellen, maar wel naar voren komt bij het IDM. Een precies waarschijnlijkheidsmodel zal besluiten dat de kans op een bepaalde gebeurtenis 0,5 is als deze gebeurtenis optrad bij de helft van de waarnemingen, ongeacht of het totale aantal waarnemingen 2 of 100 was. Een imprecies model maakt wel een onderscheid tussen deze twee situaties.

Voor het transitiemodel geeft het IDM ons voor elke $x \in \mathcal{X}$ en $y \in \mathcal{X}$ de onderstaande analoge uitdrukkingen:

$$\underline{Q}(\{x\}|y) = \frac{n_{x|y}}{\sum_{z \in \mathcal{X}} n_{z|y} + s} \text{ en } \bar{Q}(\{x\}|y) = \frac{n_{x|y} + s}{\sum_{z \in \mathcal{X}} n_{z|y} + s}.$$

De numerieke waarden van deze vergelijkingen zijn voor elke combinatie van $x \in \mathcal{X}$ en $y \in \mathcal{X}$ terug te vinden in bijlage B.6.2.

Tot slot geldt voor het emissiemodel voor alle $x \in \mathcal{X}$ en $o \in \mathcal{O}$ dat

$$\underline{S}(\{o\}|x) = \frac{m_{o|x}}{\sum_{z \in \mathcal{X}} m_{z|x} + s} \text{ en } \bar{S}(\{o\}|x) = \frac{m_{o|x} + s}{\sum_{z \in \mathcal{X}} m_{z|x} + s}.$$

De resultaten hiervan zijn voor elke combinatie van $x \in \mathcal{X}$ en $o \in \mathcal{O}$ terug te vinden in bijlage B.6.3.

We vermelden tot slot dat we er in deze toepassing voor kiezen om de voor het algoritme vereiste drempelwaarden te berekenen met vergelijking (6.24), waardoor de hierboven bepaalde onder- en bovenwaarschijnlijkheden de enige vereiste modelgegevens zijn om het EstiHMM-algoritme te kunnen uitvoeren.

8.6.2 Resultaten van het EstiHMM-algoritme

Net zoals we met het Viterbi-algoritme en de precieze modellen uit paragraaf 8.5.1 woorden konden verbeteren, kunnen we dit ook doen met het EstiHMM-algoritme en de imprecieze modellen uit de vorige paragraaf. De resultaten van deze berekening zijn terug te vinden in bijlage B.11.

Het Viterbi-algoritme geeft in de praktijk meestal één enkele oplossing en in ons voorbeeld bleek dit zelfs altijd het geval te zijn. Het EstiHMM-algoritme geeft daarentegen regelmatig verschillende mogelijke oplossingen, namelijk in die gevallen waarin de imprecieze informatie waarover we beschikken ontoereikend is om tot één unieke oplossing te komen. We geven hetzelfde voorbeeld als in paragraaf 8.5.2:

QUANTO QUANTO CUANTO DUANTO FUANTO **QUANTO**

Zoals we reeds vermeldden, slaagt het Viterbi-algoritme er in dit geval niet in om de door het OCR-programma gemaakte fout te corrigeren. Het EstiHMM-algoritme daarentegen, suggereert het correcte woord wel als één van de mogelijke oplossingen. Dit is natuurlijk enkel nuttig als we er ook daadwerkelijk in slagen om uit de verschillende gesuggereerde oplossingen het juiste te selecteren. We kunnen dit bijvoorbeeld doen door een computer te laten kijken welke van de gesuggereerde woorden in het woordenboek voorkomt, of door een mens op basis van gezond verstand te laten kiezen tussen de verschillende oplossingen. Dit vergt extra tijd, wat een nadeel is ten opzichte van het Viterbi-algoritme, maar we zullen verderop in deze paragraaf merken dat hier ook heel wat voordelen tegenover staan. Bovendien moet dit extra werk enkel uitgevoerd worden in die gevallen waar er daadwerkelijk getwijfeld wordt tussen verschillende mogelijkheden.

De resultaten van het EstiHMM-algoritme worden samengevat in de onderstaande tabel:

EstiHMM	TOTAAL	juist na OCR	fout na OCR
TOTAAL	200 (100%)	137 (68,5%)	63 (31,5%)
het juiste er bij na EstiHMM	172 (86%)	137	35
het juiste er niet bij na EstiHMM	28 (14%)	0	28

Een eerste conclusie die we uit deze tabel kunnen trekken is dat het EstiHMM-algoritme er in 86% van de gevallen in slaagt om het oorspronkelijke correcte woord te suggereren als één van de mogelijke oplossingen. Als we er van uit gaan dat we steeds de correcte oplossing kunnen selecteren uit de door het EstiHMM-algoritme gesuggereerde moge-

lijkheden, dan kunnen we het percentage correcte woorden dus opdrijven van 68,5% naar 86%. Het percentage correcte woorden na het Viterbi-algoritme bedroeg 78,5%. We zien dus dat het extra werk dat het gebruiken van het EstiHMM-algoritme met zich meebrengt er voor zorgt dat we meer fouten kunnen corrigeren.

De tabel onderscheidt – net als bij de resultaten van het Viterbi-algoritme – opnieuw vier verschillende categorieën. We geven voor elke van deze categorieën enkele voorbeelden ter illustratie en bespreken kort het verschil met de resultaten van het Viterbi-algoritme.

Er werden in totaal 63 woorden foutief ingelezen door het OCR-programma. Hiervan waren er 35 waarvoor het EstiHMM-algoritme er in slaagde om het oorspronkelijke, correcte woord te suggereren als één van de mogelijke oplossingen. We geven enkele voorbeelden:

CHE	OHS	CHE CHS
LA	LR	LA LR
TRATTAR	TAATTAR	TAATTAR TANTTAR <u>TARTTAR</u> TAZTTAR TAZZTAR TAZZZAR TRATTAR TUATTAR TZATTAR
IO	ZO	IO <u>LO</u>
TANTO	TRNTO	TANTO <u>TRNTO</u>

We merken in de bovenstaande voorbeelden dat er zich twee verschillende situaties voordoen. In de eerste twee gevallen slaagt het Viterbi-algoritme er met zijn ene suggestie meteen in om de fout te corrigeren en biedt het EstiHMM-algoritme geen meerwaarde. In tegendeel, want bij het EstiHMM-algoritme moet je achteraf nog het juiste woord kunnen selecteren uit de verschillende gesuggereerde oplossingen. In de laatste drie voorbeelden biedt het EstiHMM-algoritme dan weer wel een grote meerwaarde. Het Viterbi-algoritme slaagt er daar niet in om de fout te corrigeren en het EstiHMM-algoritme suggereert de juiste oplossing wel als mogelijkheid. Als we die juiste oplossing kunnen herkennen in de lijst met suggesties, dan zorgt het EstiHMM-algoritme er voor dat we een extra fout kunnen corrigeren.

Van de 63 woorden die foutief zijn ingelezen door het OCR-programma, waren er 28 waarvoor het EstiHMM-algoritme er niet in slaagde om deze fouten te corrigeren. Dit was bijvoorbeeld het geval bij de onderstaande woorden.

NOSTRA	NOSTRR	<u>NOSTRR</u>
OSCURA	DSCQRA	<u>DECORA</u> DEZQRA DEZZRA DSCHRA DSCORA DSCQRA DSCQZA DSCZRA DSCZZA DSZQRA DSZZRA DSZZZA DZCORA DZCQRA DZZBRA DZZFRA DZZGRA DZZORA DZZQRA DZZQUA DZZQZA DZZZRA DZZZZA DZZZZZ QSCORA QZZQRA QZZZRA ZZZQRA ZZZZRA
OGNI	DGNI	DGNI <u>DONI</u> DZZI

COME	DOME	<u>DOME</u>
USCITO	USCLTO	QZZLTO QZZZTO <u>USOLTO</u> UZZLTO UZZZTO

Aangezien de oplossing van het Viterbi-algoritme steeds vervat zit in die van het EstiHMM-algoritme, zal geen van beide algoritmes er in deze situaties in slagen om een foutief woord te corrigeren. Het EstiHMM-algoritme biedt hier dus geen meerwaarde.

Een situatie waar de voordelen van het EstiHMM-algoritme wel duidelijk naar voor komen, is als het OCR-programma een woord correct heeft ingelezen en er dus geen verbetering nodig is. Het Viterbi-algoritme dacht voor vijf van dergelijke woorden dat er een fout was opgetreden, corrigeerde die onterecht en voerde op die manier extra fouten in:

MEZZO	MEZZO	BEZZO BEZZZ BZZZO DEZZO FEZZO FEZZZ FZZZO HEZZO MBZZD MBZZO MBZZZ <u>MEZIO</u> MEZZB MEZZC MEZZD MEZZF MEZZG MEZZO MEZZQ MEZZZ MFZZO MQZZO MZZIO MZZZB MZZZC MZZZD MZZZG MZZZO MZZZQ MZZZZ NEZZO NEZZZ NZZZO PEZZO PEZZZ QZZZO QZZZZ SEZZO TEZZO VEZZO VEZZZ ZZZZO
EH	EH	CH EH <u>EN</u>
HO	HO	HO <u>NO</u>
AFFANNATA	AFFANNATA	ABFANNATA ABFANZATA ABZANNATA AFBANNATA AFFAGNATA AFFANDATA AFFANFATA AFFANGATA AFFANMATA AFFANNAIA AFFANNALA AFFANNARA AFFANNATA AFFANNAZA AFFANNTTA AFFANNZTA AFFANNZZA AFFANTATA AFFANZATA AFFANZZTA AFFANZZZA AFFAZNATA AFFAZZATA AFFZNNATA AFFZZNATA AFFZZZATA AFPANNATA AFTANNATA AFUANNATA AFZANNATA AFZANZATA AFZZZZATA AGFANNATA AGFANZATA AGZANNATA ALFANNATA AMFANNATA ANFANNATA ANTANNATA ANZANNATA APFANNATA AQFANNATA ARFANNATA ATFANNATA ATFANZATA <u>ATTANNATA</u> ATZANNATA AVFANNATA AZBANNATA AZFAGNATA AZFANDATA AZFANFATA AZFANMATA AZFANNATA AZFANNAZA AZFANNZTA . . .

ACQUA	ACQUA	AAQUA	ABBUA	ABBZA	ABDUA	ABFUA
		ABGUA	ABQBA	ABQFA	ABQQA	ABQUA
		ABQUB	ABQUC	ABQUD	ABQUE	ABQUF
		ABQUG	ABQUH	ABQUI	ABQUM	ABQUN
		ABQUO	ABQUQ	ABQUR	ABQUS	ABQUT
		ABQUV	ABQUZ	ABQVA	ABQZA	ABQZZ
		ABZUA	ABZZA	ABZZZ	ACAU A	ACBUA
		ACBZA	ACCUA	ACDUA	ACFUA	ACFZA
		ACGUA	ACGZA	ACHBA	ACHFA	ACHQA
		ACHUA	ACHVA	ACHZA	ACHZZ	ACIUA
		ACOCA	ACOLA	ACOMA	ACORA	ACOUA
		<u>ACOVA</u>	ACOZA	ACQBA	ACQUA	ACQFA
		ACQMA	ACQPA	ACQQA	ACQUA	ACQUB
		ACQUC	ACQUD	ACQUE	ACQUF	ACQUG
		ACQUH	ACQUI	ACQUL	ACQUM	ACQUN
		ACQUO	ACQUP	ACQUQ	ACQUR	...

Voor het woord AFFANNATA en ACQUA hebben we niet alle resultaten weergegeven. Een volledige lijst is terug te vinden in bijlage B.11. We zien dat het EstiHMM-algoritme er in elk van de vijf bovenstaande gevallen – in tegenstelling tot het Viterbi-algoritme – wel in slaagt om het correcte woord te suggereren als één van de mogelijke oplossingen.

Van de 137 woorden die correct ingelezen zijn door het OCR-programma, is er dus geen enkel waarvoor het EstiHMM-algoritme niet de juiste oplossing als één van de mogelijkheden suggereert. Voor elk van die 137 woorden bevat de oplossingenverzameling van het EstiHMM-algoritme het correcte woord. Behalve de vijf bovenstaande woorden, waarbij het Viterbi-algoritme in de fout ging, zijn er ook nog andere voorbeelden, waarbij het Viterbi algoritme geen bijkomende fouten introduceert:

NEL	NEL	<u>NEL</u>
VITA	VITA	<u>VITA</u>
SMARRITA	SMARRITA	<u>SMARRITA</u>
GUARDAI	GUARDAI	<u>GUARDAI</u> QUARDAI
PASSO	PASSO	<u>PASSO</u>
CHE	CHE	<u>CHE</u>

8.6.3 Verklaring van vreemde oplossingen

Bij het doornemen van de in dit hoofdstuk gegeven voorbeelden, merkt de aandachtige lezer ongetwijfeld enkele op het eerste zicht zeer onlogische oplossingen van het EstiHMM-algoritme op. Eén van de meest frappante vind ik persoonlijk de suggestie van het “woord” DZZZZZ als mogelijke correctie van het foutief als DSCQRA ingelezen woord OSCURA. We bespreken kort wat de oorzaak is van deze vreemde oplossing. Ook voor andere vreemde oplossingen kan men op dezelfde wijze op zoek

gaan naar verklaringen, maar we beperken ons hier, bij wijze van voorbeeld, tot het woord *DZZZZZ*.

Er zijn twee hoofdredenen waarom een woord als *DZZZZZ* als mogelijke oplossing wordt gesuggereerd.

Een eerste reden is dat het in de Italiaanse taal niet vreemd is om twee letters *Z* na elkaar te hebben in een woord. We zien in bijlage B.5.2 bijvoorbeeld dat in de tekst waarmee we ons model hebben opgebouwd, een *Z* slechts door drie andere letters kon gevolgd worden: een *A*, een *I* of een *Z*. De kans dat er een *Z* in een woord staat is klein, maar als dat gebeurt, dan is de kans dat er onmiddellijk daarna een tweede *Z* volgt relatief groot. Op dit punt zitten de menselijke intuïtie en de formele logica van een computer nog op dezelfde golflengte. Het probleem is nu dat de lokale modellen die we gebruiken voor een letter – door de Markovvoorwaarde – enkel afhankelijk zijn van de vorige letter. Als we bijvoorbeeld weten dat de tweede letter van een woord een *Z* is, dan is de kans dat de derde letter van dat woord een *Z* is niet afhankelijk van de waarde van de eerste letter van het woord. Dit is de reden waarom ons model het niet vreemd vindt dat een woord drie achtereenvolgende letters *Z* bevat. Als mens weten we natuurlijk dat dit onmogelijk is en dus komt een dergelijke oplossing heel absurd over. Een mogelijke – maar wiskundig niet eenvoudige – oplossing voor dit probleem is om een complexer model op te bouwen, waarbij lokale modellen niet enkel van de vorige letter, maar ook van de letter dáárvoor afhangen.

Een tweede reden waarom een woord als *DZZZZZ* als oplossing wordt gesuggereerd, heeft – in tegenstelling tot de vorige, die ook in het precieze geval geldt – te maken met onze onzekerheid over de letter *Z*. In de volledige modeltekst, die 4019 letters bevat, kwam slechts 10 keer een *Z* voor. In 2 van deze 10 gevallen was de onmiddellijk daarop volgende letter ook een *Z*. We zien in bijlage B.6.2 dat het betreffende lokale precies model hieruit concludeert dat de kans dat een *Z* door een *Z* gevolgd wordt, 20% is. Het imprecies Dirichletmodel besluit echter dat we met een dergelijke kleine hoeveelheid gegevens geen al te voorbarige conclusies mogen trekken en geeft daarom een boven- en onderwaarschijnlijkheid op die respectievelijk 16,7% en 33,3% bedraagt. Het EstiHMM-algoritme houdt vervolgens met alle waarden in dit interval rekening en beschouwt dus de mogelijkheid dat de kans dat *Z* door *Z* gevolgd wordt groter is dan 20%.

Nu we weten waarom het logisch is dat het EstiHMM-algoritme *DZZZZZ* als oplossing suggereert, is het nuttig om even stil te staan bij de reden waarom wij die oplossing als onlogisch ervaren. Dat is natuurlijk omdat wij “weten” dat *DZZZZZ* geen woord kan zijn. Het is belangrijk om te beseffen dat we hierbij uitgaan van allerlei extra informatie die niet af te leiden valt uit de data waarmee we ons model opbouwen. In vele gevallen kan die extra informatie zelfs niet of moeilijk op een formele manier omschreven worden en heeft het vooral te maken met onze eigen ervaring. Stel u een buitenaards wezen voor dat geen enkele notie heeft van de Italiaanse taal, of zelfs van taal in het algemeen. Als we dit wezen vragen om op basis van de gegevens in bijlage B.5 op zoek

te gaan naar fouten in de 200 woorden van onze voorbeeldtekst, dan zal hij sommige suggesties van het EstiHMM-algoritme als veel minder vreemd ervaren dan wij dat doen, omdat hij niet beschikt over de voorkennis die wij impliciet gebruiken.

Voor deze concrete toepassing – zijnde het verbeteren van woorden – is het verschil tussen het “taalgevoel” van een computermodel en van een mens zeer groot. Hierdoor hebben we soms de neiging om de oplossingen van het computermodel als dom of onlogisch te ervaren. In andere toepassingen, waar de enige beschikbare gegevens een berg getalletjes zijn, hebben wij als mens vaak veel minder voorkennis om op terug te vallen. In die gevallen zijn het eerder wij die dom zijn, in plaats van de computer en zullen we de oplossingen van de computer als veel nuttiger ervaren.

8.6.4 Robuustheid van het EstiHMM-algoritme

Dat het EstiHMM-algoritme vaker de correcte oplossing suggereert dan het Viterbi-algoritme, is op zich geen verdienste. Hoe meer oplossingen je suggereert, hoe vaker de correcte oplossing er één van zal zijn. Dat is altijd het geval, zelfs al geef je simpelweg tien random extra mogelijke oplossingen. Het geven van extra oplossingen is pas een verdienste als dit op een doordachte manier gebeurt. We onderzoeken daarom in deze paragraaf wanneer het EstiHMM-algoritme al dan niet meerdere oplossingen geeft en kijken welke resultaten het Viterbi-algoritme in die gevallen geeft.

We delen de 200 woorden die we hebben proberen verbeteren met het Viterbi-algoritme en het EstiHMM-algoritme op in twee groepen. Een eerste groep bestaat uit de 155 woorden waarvoor het EstiHMM-algoritme slechts één enkele oplossing gaf. De tweede groep zijn de overige 45 woorden, waarvoor het EstiHMM-algoritme dus meerdere mogelijkheden suggereerde als oplossing.

We toonden eerder reeds aan (zie paragraaf 6.3) dat de oplossing van het Viterbi-algoritme altijd één van de oplossingen van het EstiHMM-algoritme is. Als het EstiHMM-algoritme slechts één oplossing geeft, dan volgt hier dus uit dat het Viterbi-algoritme voor dat woord tot dezelfde oplossing komt. In deze gevallen is er dus geen verschil tussen beide algoritmes. We kunnen daarom voor de 155 woorden waarbij het EstiHMM-algoritme één oplossing geeft, de resultaten van Viterbi-algoritme en het EstiHMM-algoritme samen bespreken, aangezien ze identiek zijn. De resultaten zijn samengevat in de onderstaande tabel.

EstiHMM geeft één oplossing	TOTAAL	juist na OCR	fout na OCR
TOTAAL	155 (100%)	129 (83,2%)	26 (16,8%)
juist na EstiHMM/Viterbi	134 (86,5%)	129	5
fout na EstiHMM/Viterbi	21 (13,5%)	0	21

Het percentage woorden dat door het OCR-programma correct wordt ingelezen bedraagt hier 83,2%. Als we dit vergelijken met het globale percentage correct ingelezen woorden,

namelijk 68,5%, dan kunnen we hieruit afleiden dat het EstiHMM-algoritme voor correct ingelezen woorden vaker één oplossing aanreikt dan voor woorden die fouten bevatten. We zien ook dat het percentage correcte woorden na het uitvoeren van de algoritmes 86,5% bedraagt, wat maar een lichte verbetering is ten opzichte van de 83,2% woorden die al correct waren.

Voor de 45 woorden waarbij het EstiHMM-algoritme twijfelt, en dus meerdere oplossingen suggereert, is er wel een duidelijk verschil tussen de resultaten van het Viterbi-algoritme en het EstiHMM-algoritme. Het Viterbi-algoritme geeft namelijk telkens maar één oplossing. De resultaten van beide algoritmes zijn voor dit geval samengevat in de onderstaande tabel.

EstiHMM twijfelt	TOTAAL	juist na OCR	fout na OCR
TOTAAL	45 (100%)	8 (17,8%)	37 (82,2%)
juist na Viterbi	23 (51,1%)	3	20
fout na Viterbi	22 (48,9%)	5	17
het juiste er bij na EstiHMM	38 (84,4%)	8	30
het juiste er niet bij na EstiHMM	7 (15,6%)	0	7

Een eerste, zeer opvallend iets dat we kunnen afleiden uit deze tabel, is dat het EstiHMM-algoritme veel vaker twijfelt bij woorden die foutief zijn ingelezen door het OCR-programma, dan bij woorden die correct zijn ingelezen. Als het EstiHMM-algoritme twijfelt, zien we dat het beschouwde woord in slechts in 17,8% van de gevallen correct werd ingelezen. Men kan het al dan niet twijfelen van het EstiHMM-algoritme dus gebruiken als indicatie van de correctheid van het woord dat verbeterd wordt.

Een tweede conclusie is dat in die gevallen waar het EstiHMM-algoritme twijfelt, na afloop van het Viterbi-algoritme slechts 51,1% van de woorden correct zijn. Dit staat in schril contrast met het globale percentage correcte woorden na afloop van het Viterbi-algoritme, waarvan we in paragraaf 8.5.2 leerden dat dit 78,5% bedraagt. Het EstiHMM-algoritme daarentegen geeft in 84,4% van de gevallen het correcte resultaat als één van de mogelijkheden, wat zeer goed vergelijkbaar is met het globale percentage, dat 86% bedraagt. Het EstiHMM-algoritme slaagt er dus in om in die gevallen waar het Viterbi-algoritme het moeilijk heeft en meer fouten maakt, toe te geven dat het moeilijk wordt en meerdere oplossingen te suggereren waartussen het algoritme twijfelt.

We kunnen uit de analyses in deze paragraaf besluiten dat het EstiHMM-algoritme een manier is om de resultaten van het Viterbi-algoritme meer robuust te maken. Als het EstiHMM-algoritme meerdere resultaten geeft, is dit namelijk een indicatie dat er robuustheidsproblemen optreden bij de oplossing die bekomen wordt door een precies model dat op dezelfde data is gebaseerd. Onze aanpak vermijdt dat er voor dergelijke moeilijke problemen “gegokt” wordt en geeft verschillende mogelijke oplossingen waartussen getwijfeld wordt. Als men een dergelijk robuustheidsprobleem – dankzij het EstiHMM-algoritme – opmerkt, heeft men de keuze tussen twee manieren om hier

mee om te gaan. Een eerste mogelijkheid is om met andere criteria, bijvoorbeeld op basis van menselijke ervaring, een gefundeerde keuze te proberen maken tussen de verschillende gesuggereerde oplossingen, waardoor het uiteindelijke aantal correcte resultaten stijgt. Een andere mogelijkheid is concluderen dat men over te weinig data beschikt. Vervolgens kan men beslissen om extra data te verzamelen en op die manier het robuustheidsprobleem oplossen.

8.6.5 Analyse van het aantal oplossingen

We besteden in deze paragraaf nog even aandacht aan het aantal oplossingen dat door het EstiHMM-algoritme wordt gesuggereerd. Stel u het fictieve IKWEETVANNIETS-algoritme voor dat voor elk te verbeteren woord alle mogelijke woorden met diezelfde lengte als oplossing suggereert. Dit algoritme is zeer robuust en heeft bovendien het grote voordeel dat de correcte oplossing altijd één van de gesuggereerde woorden zal zijn. Toch vermoed ik dat u niet geneigd zal zijn om dit algoritme vaak te gebruiken. De reden is natuurlijk dat de robuustheid die gepaard gaat met het aanbieden van meerdere mogelijke oplossingen, slechts een praktisch nut heeft als dit aantal mogelijke oplossingen beperkt blijft.

Voor de woorden waar het EstiHMM-algoritme maar één oplossing suggereert, heeft het weinig zin om het aantal oplossingen te analyseren. We beperken ons dus tot de 45 woorden waarvoor het EstiHMM-algoritme twijfelt. We maken verder ook een onderscheid tussen twee verschillende gevallen, namelijk deze waarbij het Viterbi-algoritme er in slaagde om het correcte woord als oplossing te geven en deze waarbij het Viterbi-algoritme een foutief woord als oplossing gaf.

We beschouwen eerst de 23 woorden waarvoor het Viterbi-algoritme de correcte oplossing geeft en het EstiHMM-algoritme twijfelt. Het EstiHMM-algoritme zal dan per definitie de correcte oplossing ook als één van de mogelijkheden suggereren, maar heeft als nadeel dat er ook andere oplossingen gesuggereerd worden. We geven enkele voorbeelden:

CHE	OHS	CHE CHS
LA	LR	LA LR
SELVA	SFLVH	SELVA SELVH SELZH SFLVH SFZVH SFZZH SFZZZ SZLVH SZZVH SZZZH SZZZZ
LA	ZA	LA ZA
SONNO	SGNNO	SGNNO SONNO
DOVE	DCVE	DOVE DZVE
TERMINAVA	TERMLNAVA	TERALNAVA TERALZAVA TERBZZAVA TERMINAVA TERMZNAVA TERMZZAVA TERMZZZVA TERMZZZZA TERNZNAVA

		TERNZZAVA	TERNZZZVA	TERNZZZZA	
		TERQZZAVA	TERZGNAVA	TERZINAVA	
		TERZLNAVA	TERZLZAVA	TERZLZZVA	
		TERZZFAVA	TERZZMAVA	TERZZNAVA	
		TERZZNAZA	TERZZNZVA	TERZZNZZA	
		TERZZZAVA	TERZZZAZA	TERZZZZIA	
		TERZZZZVA	TERZZZZZA	TERZZZZZZ	
		TEZMINAVA	TEZMZZAVA	TEZZZMAVA	
		TEZZZNAVA	TEZZZNZVA	TEZZZZAVA	
		TEZZZZAZA	TEZZZZZVA	TEZZZZZZA	
		TZZZZZAVA			
GUARDAI	GUARDAI	<u>GUARDAI</u>	QUARDAI		
CHETA	CHRTA	CHATA	<u>CHETA</u>	CHRTA	CHZTA CHZZA
CON	UON	<u>CON</u>	UON		

Voor de praktische bruikbaarheid van het EstiHMM-algoritme ten opzichte van het Viterbi-algoritme, is het belangrijk dat het aantal oplossingen dat het EstiHMM-algoritme in bovenstaande gevallen suggereert beperkt blijft. Dit blijkt inderdaad het geval te zijn. In 12 van de 23 gevallen geeft het EstiHMM-algoritme slechts twee mogelijke oplossingen. In de overige gevallen blijft het aantal oplossingen meestal kleiner dan tien en het grootste aantal oplossingen dat gesuggereerd wordt is 40, namelijk voor het woord TERMINAVA. Gemiddeld geeft het EstiHMM-algoritme voor de hier beschouwde woorden 7,4 mogelijke oplossingen.

Merk op dat we hier enkel woorden bekeken hebben waarvoor het EstiHMM-algoritme twijfelt. Als we de volledige verzameling beschouwen van alle 157 woorden waarvoor het Viterbi-algoritme de correcte oplossing geeft, dan is het gemiddelde aantal door het EstiHMM-algoritme gesuggereerde oplossingen natuurlijk veel lager. In dat geval worden gemiddeld slechts 1,94 oplossingen voorgesteld per woord.

We bekijken tot slot de 22 woorden waarvoor het EstiHMM-algoritme twijfelde en het Viterbi-algoritme de correcte oplossing niet gaf. In die gevallen is het minder belangrijk dat het aantal oplossingen van het EstiHMM-algoritme beperkt blijft, aangezien het Viterbi-algoritme sowieso de correcte oplossing niet vindt. Het is in tegendeel zelfs heel normaal dat het EstiHMM-algoritme veel oplossingen suggereert, aangezien dit typisch de gevallen zijn waarbij het Viterbi-algoritme met robuustheidsproblemen kampt. We geven opnieuw enkele voorbeelden:

OSCURA	DSCQRA	<u>DECORA</u>	DEZQRA	DEZZRA	DSCHRA	DSCORA
		DSCQRA	DSCQZA	DSCZRA	DSCZZA	DSZQRA
		DSZZRA	DSZZZA	DZCORA	DZCQRA	DZZBRA
		DZZFRA	DZZGRA	DZZORA	DZZQRA	DZZQUA
		DZZQZA	DZZZRA	DZZZZA	DZZZZZ	QSCORA
		QZZQRA	QZZZRA	ZZZQRA	ZZZZRA	
QUANTO	OUANTO	CUANTO	<u>DUANTO</u>	FUANTO	QUANTO	

TRATTAR	TAATTAR	TAATTAR TANTTAR <u>TARTTAR</u> TAZTTAR TAZZTAR TAZZZAR TRATTAR TUATTAR TZATTAR
ABBANDONAI	ABBANDONAZ	<u>ABBANDONAL</u> ABBANDONAZ ABBANDONZZ ABZANDONAL ABZANDONAZ ABZANDONZZ AZBANDONAL AZBANDONAZ AZBANDONZZ AZZANDONAL AZZANDONAZ AZZANDONZZ BBBANDONAL BBBANDONAZ BBBANDONZZ FBBANDONAL FBBANDONAZ FBBANDONZZ MBBANDONAL MBBANDONAZ MBBANDONZZ QBBANDONAL QBBANDONAZ QBBANDONZZ ZBBANDONZZ
ACQUA	ACQUA	AAQUA ABBUA ABBZA ABDUA ABFUA ABGUA ABQBA ABQFA ABQQA ABQUA ABQUB ABQUC ABQUD ABQUE ABQUF ABQUG ABQUH ABQUI ABQUM ABQUN ABQUO ABQUQ ABQUR ABQUS ABQUT ABQUV ABQUZ ABQVA ABQZA ABQZZ ABZUA ABZZA ABZZZ ACAUA ACBUA ACBZA ACCUA ACDUA ACFUA ACFZA ACGUA ACGZA ACHBA ACHFA ACHQA ACHUA ACHVA ACHZA ACHZZ ACIUA ACOCA ACOLA ACOMA ACORA ACOUA <u>ACOVA</u> ACOZA ACQBA ACQCA ACQFA ACQMA ACQPA ACQQA ACQUA ACQUB ACQUC ACQUD ACQUE ACQUF ACQUG ACQUH ACQUI ACQUL ACQUM ACQUN ACQUO ACQUP ACQUQ ACQUR ...
VIVA	VIVR	VIBR VIOR <u>VIUR</u> VIVA VIVR VIZR VZZR

Deze voorbeelden doen ons vermoeden dat het aantal oplossingen dat het EstiHMM-algoritme suggereert, in die gevallen waar het Viterbi-algoritme de correcte oplossing niet vindt, relatief groot is. Dit blijkt inderdaad het geval te zijn. Het grootste aantal oplossingen vinden we voor het woord ACQUA, namelijk 823. Gemiddeld worden er 66 oplossingen voorgesteld. Dit gemiddelde wordt weliswaar beïnvloed door extreme voorbeelden als het woord ACQUA, maar toch blijkt duidelijk dat het EstiHMM-algoritme een groot aantal oplossingen geeft in die situaties waar het Viterbi-algoritme de correcte oplossing niet vindt. Dit bevestigt nogmaals dat de oplossingen van het EstiHMM-algoritme gebruikt kunnen worden als een indicator van de robuustheid van het Viterbi-algoritme.

Hoofdstuk 9

Besluit

“Wat de rups het einde noemt, noemt de rest van de wereld een vlinder.”

– Lao Tse

Door de grafische structuur van een imprecies hidden Markovmodel te interpreteren met de onafhankelijkheidsnotie van epistemische irrelevantie zijn we in staat om een efficiënt algoritme te ontwikkelen dat de maximale sequenties van dit model bepaalt. Dit EstiHMM-algoritme veralgemeent daarmee het gekende Viterbi-algoritme, dat hetzelfde probleem oplost in het precieze geval.

Initiële experimenten en een eerste toepassing tonen aan dat het EstiHMM-algoritme er in slaagt eventuele robuustheidsproblemen van het Viterbi-algoritme aan te geven en op te vangen. In die gevallen waar het Viterbi-algoritme met robuustheidsproblemen kampt, zal het EstiHMM-algoritme namelijk meerdere oplossingen suggereren. Dit is een indicatie dat we over te weinig data beschikken om op betrouwbare wijze tot één enkele, precieze oplossing te komen. In situaties waar een model moet opgebouwd worden aan de hand van een beperkte hoeveelheid data, kan een dergelijke robuuste aanpak volgens mij een grote meerwaarde bieden.

Ik hoop dan ook dat het EstiHMM-algoritme zijn weg mag vinden buiten deze masterproef en nuttig gebruikt kan worden in de vele toepassingsdomeinen waar het Viterbi-algoritme nu reeds lang zijn dienst bewijst.

Ik eindig dit besluit graag met het opsommen van enkele verdere onderzoeksmogelijkheden. De onderstaande voorstellen lijken me interessante uitdagingen:

- Het uitvoeren van bijkomende experimenten met het EstiHMM-algoritme en het zoeken naar en uitwerken van concrete toepassingen.

-
- Het ontwerpen van een algoritme dat de maximale sequenties van een hidden Markovmodel kan bepalen als we de grafische structuur hiervan interpreteren aan de hand van epistemische onafhankelijkheid, in plaats van de in deze masterproef beschouwde epistemische irrelevantie.
 - Het ontwikkelen van bijkomende, theoretisch goed onderbouwde methoden om imprecieze modellen op te bouwen in die gevallen waar de verborgen toestanden niet of moeilijk kunnen waargenomen worden.
 - Het uitbreiden van de in het kader van het EstiHMM-algoritme ontwikkelde principes, zodat ze kunnen toegepast worden in meer algemene credale netwerken dan hidden Markovmodellen.

Ik kijk er naar uit om deze boeiende pistes verder te bewandelen in het kader van een doctoraatsonderzoek.

Bijlage A

Bewijzen

“For those who believe, no proof is necessary. For those who don’t believe, no proof is possible.”

– Stuart Chase

Bewijs van eigenschap 4.1

We bewijzen de verschillende ongelijkheden $\min f \leq \underline{P}(f) \leq \overline{P}(f) \leq \max f$ één voor één. De eerste ongelijkheid is een rechtstreekse toepassing van axioma C1. Uit axioma C3 volgt dat $\underline{P}(f - f) \geq \underline{P}(f) + \underline{P}(-f)$, en aangezien we door axioma C2 weten dat $\underline{P}(0) = 0$ geldt dus dat $\underline{P}(f) \leq -\underline{P}(-f) = \overline{P}(f)$, waarbij de laatste gelijkheid volgt uit definitie (4.5). Als we axioma C1 toepassen voor $-f$, dan vinden we dat $\min -f \leq \underline{P}(-f)$ en dus dat $\max f = -\min -f \geq -\underline{P}(-f) = \overline{P}(f)$.

We bewijzen tot slot dat $\underline{P}(f) = \overline{P}(\mu) = \mu$ door bovenstaande toe te passen voor $f = \mu$. We zien dat $\mu = \min \mu \leq \underline{P}(\mu) \leq \overline{P}(\mu) \leq \max \mu = \mu$, waaruit het gestelde onmiddellijk volgt. \square

Bewijs van eigenschap 4.2

Voor alle $x \in \mathcal{X}$ is $f(x) > g(x)$ en dus $f(x) - g(x) > 0$. Aangezien alle functiewaarden van $f - g$ strikt positief zijn, weten we dat $0 < \min(f - g) \leq \underline{P}(f - g)$, waarbij de laatste ongelijkheid volgt uit axioma C1. Door axioma C3 weten we bovendien dat $\underline{P}(f) = \underline{P}((f - g) + g) \geq \underline{P}(f - g) + \underline{P}(g)$ en dus dat $\underline{P}(f) - \underline{P}(g) \geq \underline{P}(f - g) > 0$, waaruit tot slot volgt dat $\underline{P}(f) > \underline{P}(g)$. \square

Bewijs van eigenschap 4.3

Het bewijs verloopt volledig analoog aan dat van eigenschap 4.2, met dat verschil dit keer $f(x) - g(x) \geq 0$ voor alle $x \in \mathcal{X}$, waardoor $0 \leq \min(f - g) \leq \underline{P}(f - g)$. \square

Bewijs van eigenschap 4.4

Fixeer een natuurlijk getal $n \geq 1$, $f_0, \dots, f_n \in \mathcal{L}(\mathcal{X})$ en $\lambda_0, \dots, \lambda_n \geq 0$. Stel $g_0 = \lambda_0 f_0$ en $g = \sum_{j=1}^n \lambda_j f_j$ en voer $h = g_0 - g$ in. Aangezien $\mathcal{L}(\mathcal{X})$ een lineaire ruimte is, weten

we dat $g_0, g, h \in \mathcal{L}(\mathcal{X})$. Uit axioma C3 volgt dat $\underline{P}(g_0) = \underline{P}(g + h) \geq \underline{P}(g) + \underline{P}(h)$. Verder weten we uit axioma C2 dat $\underline{P}(g_0) = \lambda_0 \underline{P}(f_0)$, volgt uit axioma C2 en C3 dat $\underline{P}(g) \geq \sum_{j=1}^n \lambda_j \underline{P}(f_j)$ en uit axioma C1 dat $\underline{P}(h) \geq \min h$. Uit dit alles volgt dat

$$\begin{aligned} \max(\sum_{j=1}^n \lambda_j f_j - \lambda_0 f_0) &= \max(g - g_0) = \max(-h) = -\min h \\ &\geq -\underline{P}(h) \geq \underline{P}(g) - \underline{P}(g_0) \\ &\geq \sum_{j=1}^n \lambda_j \underline{P}(f_j) - \lambda_0 \underline{P}(f_0). \end{aligned}$$

Als we tot slot voor alle $f \in \mathcal{L}(\mathcal{X})$ stellen dat $G(f) = f - \underline{P}(f)$, dan volgt uit het bovenstaande onmiddellijk dat $\max(\sum_{j=1}^n \lambda_j G(f_j) - \lambda_0 G(f_0)) \geq 0$. \square

Bewijs van eigenschap 4.5

Dat $\rho_f(\mu)$ een niet-stijgende functie van μ is, is een rechtstreeks gevolg van eigenschap 4.3.

Beschouw nu een willekeurige μ_1 en μ_2 in \mathbb{R} en een $0 \leq \lambda \leq 1$. $\rho_f(\mu)$ is concaaf omdat

$$\begin{aligned} \rho_f(\lambda \mu_1 + (1 - \lambda) \mu_2) &= \underline{P}(\mathbb{I}_{\{y\}}[f - (\lambda \mu_1 + (1 - \lambda) \mu_2)]) \\ &= \underline{P}(\lambda \mathbb{I}_{\{y\}}[f - \mu_1] + (1 - \lambda) \mathbb{I}_{\{y\}}[f - \mu_2]) \\ &\geq \underline{P}(\lambda \mathbb{I}_{\{y\}}[f - \mu_1]) + \underline{P}((1 - \lambda) \mathbb{I}_{\{y\}}[f - \mu_2]) \\ &= \lambda \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_1]) + (1 - \lambda) \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_2]) \\ &= \lambda \rho_f(\mu_1) + (1 - \lambda) \rho_f(\mu_2), \end{aligned}$$

waarbij de ongelijkheid volgt uit het coherentie-axioma C3 en de daaropvolgende gelijkheid uit coherentie-axioma C2.

Beschouw nu een μ_1 en μ_2 in \mathbb{R} , waarbij $\mu_2 > \mu_1$. Er geldt dat

$$\begin{aligned} \rho_f(\mu_2) &= \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_2]) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_1 + (\mu_1 - \mu_2)]) \\ &= \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_1] + \mathbb{I}_{\{y\}}(\mu_1 - \mu_2)) \geq \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_1]) + \underline{P}(\mathbb{I}_{\{y\}}(\mu_1 - \mu_2)) \\ &= \rho_f(\mu_1) - (\mu_2 - \mu_1) \bar{P}(\{y\}) \end{aligned}$$

waarbij de ongelijkheid volgt uit het coherentie-axioma C3 en de daaropvolgende gelijkheid uit definitie (4.5) en het coherentie-axioma C2. Aangezien $\rho_f(\mu)$ een niet-stijgende functie is kunnen we deze ongelijkheid omvormen tot

$$|\rho_f(\mu_1) - \rho_f(\mu_2)| = \rho_f(\mu_1) - \rho_f(\mu_2) \leq (\mu_2 - \mu_1) \bar{P}(\{y\}) = |\mu_2 - \mu_1| \bar{P}(\{y\}),$$

waarmee bewezen is dat $\rho_f(\mu)$ een (Lipschitz-)continue functie is. \square

Bewijs van eigenschap 4.6

Kies een $\mu^- < 0$ die tevens kleiner is dan de minimale waarde die f aanneemt binnen haar domein. Merk op dat dit minimum wegens aanname 1 goed gedefinieerd is. We vinden dat

$$\rho_f(\mu^-) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu^-]) \geq \underline{P}(\mathbb{I}_{\{y\}}[\min f - \mu^-]) = (\min f - \mu^-) \underline{P}(\{y\}) \geq 0.$$

De ongelijkheid is hierbij een toepassing van eigenschap 4.3 en de laatste gelijkheid volgt uit coherentie-axioma C2.

Kies vervolgens een $\mu^+ > 0$ die tevens groter is dan de maximale waarde die f aanneemt binnen haar domein. We vinden op analoge wijze dat

$$\rho_f(\mu^+) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu^+]) \leq \underline{P}(\mathbb{I}_{\{y\}}[\max f - \mu^+]) = -(\mu^+ - \max f)\bar{P}(\{y\}) \leq 0.$$

De ongelijkheid is opnieuw een toepassing van eigenschap 4.3 en de laatste gelijkheid volgt uit coherentie-axioma C2 en definitie (4.5). \square

Bewijs van eigenschap 4.7

Beschouw een μ_1 en μ_2 in \mathbb{R} , waarbij $\mu_2 > \mu_1$. Als $\underline{P}(\{y\}) > 0$, dan geldt dat

$$\begin{aligned} \rho_f(\mu_1) &= \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_1]) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_2 + (\mu_2 - \mu_1)]) \\ &= \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_2] + \mathbb{I}_{\{y\}}(\mu_2 - \mu_1)) \geq \underline{P}(\mathbb{I}_{\{y\}}[f - \mu_2]) + \underline{P}(\mathbb{I}_{\{y\}}(\mu_2 - \mu_1)) \\ &= \rho_f(\mu_2) + (\mu_2 - \mu_1)\underline{P}(\{y\}) > \rho_f(\mu_2) \end{aligned}$$

waarbij de eerste ongelijkheid volgt uit het coherentie-axioma C3 en de daaropvolgende gelijkheid uit het coherentie-axioma C2. Dit resultaat impliceert dat $\rho_f(\mu)$ een dalende functie is.

Uit eigenschap 4.6 volgt dat $\rho_f(\mu)$ minstens één nulpunt heeft, en aangezien $\rho_f(\mu)$ in dit geval dalend is weten we dat dit nulpunt uniek is. \square

Bewijs van eigenschap 4.8

Uit eigenschap 4.1 en de onderstelling dat $\bar{P}(\{y\}) = 0$ volgt onmiddellijk dat ook $\underline{P}(\{y\}) = 0$. Fixeer nu een μ in \mathbb{R} en kies dan een a en b in \mathbb{R} zodanig dat

$$\begin{aligned} a &< \min\{0, \min\{f - \mu\}\} \\ b &> \max\{0, \max\{f - \mu\}\}. \end{aligned}$$

We vinden dan dat

$$\rho_f(\mu) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu]) \geq \underline{P}(\mathbb{I}_{\{y\}}a) = a\bar{P}(\{y\}) = 0$$

en ook dat

$$\rho_f(\mu) = \underline{P}(\mathbb{I}_{\{y\}}[f - \mu]) \leq \underline{P}(\mathbb{I}_{\{y\}}b) = b\underline{P}(\{y\}) = 0.$$

De verantwoording van en de gebruikte eigenschappen en definities in bovenstaande afleidingen, zijn analoog aan deze die voorkwamen in het bewijs van eigenschap 4.6. Uit de hierboven afgeleide twee ongelijkheden volgt dat $\rho_f(\mu)$ gelijk is aan nul voor elke willekeurige μ in \mathbb{R} . \square

Bewijs van eigenschap 4.9

Aangezien $\rho_f(\mu)$ door eigenschap 4.5 een niet stijgende functie is, volgt uit definitie (4.12) dat $\rho_f(\mu) \geq 0$ in het interval $]-\infty, \underline{P}(f | y)]$ en dat $\rho_f(\mu) < 0$ in het interval

$]\underline{P}(f|y), +\infty[$. In het bewijs van eigenschap 4.8 toonden we reeds aan dat $\rho_f(\mu) \leq 0$ als $\underline{P}(\{y\}) = 0$, waaruit we afleiden dat $\rho_f(\mu) = 0$ in het interval $]-\infty, \underline{P}(f|y)[$. We bewijzen tot slot uit het ongerijmde dat $\rho_f(\mu)$ een dalende functie van μ is in het interval $]\underline{P}(f|y), +\infty[$. Als $\rho_f(\mu)$ niet dalend zou zijn in dat interval, dan bestaan er een μ_1 en een μ_2 in dit interval, met $\mu_2 > \mu_1$, waarvoor $0 > \rho_f(\mu_2) \geq \rho_f(\mu_1)$. Aangezien $\rho_f(\mu)$ gelijk is aan nul in het interval $]-\infty, \underline{P}(f|y)[$, bestaat er ook een $\mu_0 < \mu_1$, waarvoor $\rho_f(\mu_0) = 0$. Het bestaan van een dergelijke μ_0, μ_1 en μ_2 is echter strijdig met de concaviteit van $\rho_f(\mu)$ die volgt uit eigenschap 4.5. We besluiten dus dat $\rho_f(\mu)$ wel degelijk dalend moet zijn in $]\underline{P}(f|y), +\infty[$. \square

Bewijs van eigenschap 4.10

Uit eigenschap 4.1 volgt dat $\bar{P}(\{y\}) \geq \underline{P}(\{y\}) \geq 0$. Bijgevolg sluiten de drie gevallen die in eigenschap 4.7, 4.8 en 4.9 besproken worden elkaar onderling uit en zal bovendien telkens één van deze drie gevallen optreden. Als er nu een a bestaat waarvoor $\rho_f(a) > 0$ dan weten we dat we ons in het geval bevinden dat besproken wordt in eigenschap 4.7 en dus dat $\rho_f(\mu)$ een dalende functie is van μ . \square

Bewijs van eigenschap 4.11

In het bewijs van eigenschap 4.10 werd reeds aangetoond dat de drie gevallen die in eigenschap 4.7, 4.8 en 4.9 besproken worden elkaar onderling uitsluiten en dat er telkens één van deze drie gevallen optreedt. Als $\rho_f(\mu)$ nu in een willekeurig interval $]a, b[$ negatief is, dan bevinden we ons ofwel in het geval van eigenschap 4.7, en is $\rho_f(\mu)$ dus dalend in zijn hele domein, ofwel bevinden we ons in het geval van eigenschap 4.9, waaruit volgt dat $\rho_f(\mu)$ in het interval $]a, b[$ dalend is. \square

Bewijs van eigenschap 5.1

Als alle lokale bovenwaarschijnlijkheden positief zijn, dan volgt uit vergelijking (5.5) en (5.9) rechtstreeks dat $\bar{P}_k(\{x_{k:n}\} \times \{o_{k:n}\} | x_{k-1}) > 0$ en $\bar{P}_k(\{x_{k:n}\} | x_{k-1}) > 0$.

We weten verder ook dat $\bar{P}_k(\{o_{k:n}\} | x_{k-1}) > 0$ omdat

$$\begin{aligned} \bar{P}_k(\{o_{k:n}\} | x_{k-1}) &= \bar{P}_k(\mathbb{I}_{\{o_{k:n}\}} | x_{k-1}) \\ &= \bar{P}_k(\mathbb{I}_{\{o_{k:n}\}} \sum_{z_{k:n} \in \mathcal{X}_{k:n}} \mathbb{I}_{\{z_{k:n}\}} | x_{k-1}) \geq \bar{P}_k(\mathbb{I}_{\{o_{k:n}\}} \mathbb{I}_{\{z_{k:n}^*\}} | x_{k-1}) > 0, \end{aligned}$$

waarbij we de tweede gelijkheid kunnen schrijven omdat $\sum_{z_{k:n} \in \mathcal{X}_{k:n}} \mathbb{I}_{\{z_{k:n}\}} = 1$. De eerste ongelijkheid volgt uit eigenschap 4.3 en de tweede volgt uit formule (5.5) en de uitgangsveronderstelling dat alle lokale bovenwaarschijnlijkheden positief zijn. $z_{k:n}^*$ is in bovenstaande afleiding een willekeurig element van $\mathcal{X}_{k:n}$.

Op analoge wijze vinden we tot slot dat

$$\bar{E}_k(\{o_{k:n}\} | x_k) = \bar{E}_k(\mathbb{I}_{\{o_{k:n}\}} \sum_{z_{k+1:n} \in \mathcal{X}_{k+1:n}} \mathbb{I}_{\{z_{k+1:n}\}} | x_k) \geq \bar{E}_k(\mathbb{I}_{\{o_{k:n}\}} \mathbb{I}_{\{z_{k+1:n}^*\}} | x_k) > 0,$$

waarbij we voor de laatste ongelijkheid ditmaal formule (5.7) gebruikt hebben. \square

Bewijs van eigenschap 5.2

Als alle lokale onderwaarschijnlijkheden positief zijn, dan volgt uit vergelijking (5.4) en (5.8) rechtstreeks dat $\underline{P}_k(\{x_{k:n}\} \times \{o_{k:n}\} | x_{k-1}) > 0$ en $\underline{P}_k(\{x_{k:n}\} | x_{k-1}) > 0$.

We weten verder ook dat $\underline{P}_k(\{o_{k:n}\} | x_{k-1}) > 0$ omdat

$$\begin{aligned} \underline{P}_k(\{o_{k:n}\} | x_{k-1}) &= \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}} | x_{k-1}) \\ &= \underline{P}_k(\mathbb{I}_{\{o_{k:n}\}} \sum_{z_{k:n} \in \mathcal{X}_{k:n}} \mathbb{I}_{\{z_{k:n}\}} | x_{k-1}) \geq \sum_{z_{k:n} \in \mathcal{X}_{k:n}} \underline{P}_k(\mathbb{I}_{\{z_{k:n}\}} | x_{k-1}) > 0, \end{aligned}$$

waarbij we de tweede gelijkheid kunnen schrijven omdat $\sum_{z_{k:n} \in \mathcal{X}_{k:n}} \mathbb{I}_{\{z_{k:n}\}} = 1$. De eerste ongelijkheid volgt uit coherentie-axioma C3 en de tweede volgt uit formule (5.4) en de uitgangsveronderstelling dat alle lokale onderwaarschijnlijkheden positief zijn.

Op analoge wijze vinden we tot slot dat

$$\begin{aligned} \underline{E}_k(\{o_{k:n}\} | x_k) &= \underline{E}_k(\mathbb{I}_{\{o_{k:n}\}} \sum_{z_{k+1:n} \in \mathcal{X}_{k+1:n}} \mathbb{I}_{\{z_{k+1:n}\}} | x_k) \\ &\geq \sum_{z_{k+1:n} \in \mathcal{X}_{k+1:n}} \underline{E}_k(\mathbb{I}_{\{o_{k:n}\}} \mathbb{I}_{\{z_{k+1:n}\}} | x_k) > 0, \end{aligned}$$

waarbij we voor de laatste ongelijkheid ditmaal formule (5.6) gebruikt hebben. \square

Data Divina Commedia

“... non men che saver, dubbiar m'aggrata.”

– Dante Alighieri in *La Divina Commedia*

B.1 Originele modeltekst

Lo giorno se n'andava, e l'aer bruno
toglieva li animai che sono in terra
dalle fatiche loro; ed io sol uno

m'apparecchiava a sostener la guerra
sì del cammino e sì de la pietate,
che ritrarrà la mente che non erra.

O muse, o alto Ingegno, or m'aiutate!
O mente, che scrivesti ciò ch'io vidi,
qui si parrà la tua nobilitate!

Io cominciai: “Poeta che mi guidi,
guarda la mia virtù, s'ella è possente,
prima che all'alto passo tu mi fidi.

Tu dici che di Silvio lo parente,
corruttibile ancora, ad immortale
secolo andò, e fu sensibilmente.

Però, se l'avversario d'ogni male
cortese i fu, pensando l'alto effetto
che uscìr dovea di lui, e il chi, e il quale,

non pare indegno ad uomo d'intelletto;
ch'ei fu dell'alma Roma e di suo impero
nell'Empireo Ciel per padre eletto;

la quale e il quale, a voler dir lo vero,
fur stabiliti per lo loco santo
u'siede il successor del maggior Piero.

Per questa andata, onde gli dà tu vanto,
intese cose, che furon cagione
di sua vittoria e del Papale Ammanto.

Andovvi poi il Vaso d'Elezione,
per recarne conforto a quella fede
ch'è principio alla via di salvazione.

Ma io, perchè venirvi? o chi 'l concede?
Io non Enea, io non Paulo sono;
me degno a ciò nè io nè altri crede.

Per che, se del venire io m'abbandono,
temo che la venuta non sia folle;
se' savio, e intendi me' ch'io non ragiono”.

E quale è quei che disvuol ciò che volle
e per nuovi pensier cangia proposta
sì, che dal cominciar tutto si tolle;

tal mi fec' io in quella oscura costa;
per che, pensando, consumai l'impresa,
che fu nel cominciar cotanto tosta.

“Se io ho ben la tua parola intesa,”
rispose del Magnanimo quell’ ombra,
“l’anima tua è da viltate offesa,

la qual molte fiata l’uomo ingombra
sì, che d’onrata impresa lo rivolve,
come falso veder bestia, quand’ ombra.

Da questa tema acciò che tu ti solve,
dirotti perch’io venni, e quel che intesi
nel primo punto che di te mi dolve.

Io era tra color che son sospesi,
e Donna mi chiamò beata e bella
tal, che di comandar io la richiesi.

Lucevan gli occhi suoi più che la stella;
e cominciommi a dir soave e piana,
con angelica voce, in sua favella:

"O anima cortese Mantovana,
di cui la fama ancor nel mondo dura,
e durerà quanto il mondo lontana,

l’amico mio, e non della Ventura,
nella deserta spiaggia è impedito
sì nel cammin, che vòlto è per paura;

e temo che non sia già sì smarrito,
ch’io mi sia tardi al soccorso levata,
per quel ch’io ho di lui nel Ciel udito.

Or muovi, e con la tua parola ornata,
e con ciò ch’è mestieri al suo campare,
l’aiuta sì, ch’io ne sia consolata.

Io son Beatrice che ti faccio andare;
vegno del loco ove tornar desiò;
amor mi mosse, che mi fa parlare.

Quando sarò dinanzi al Signor mio,
di te mi loderò sovente a lui."
Tacette allora, e poi cominciò io:

"O donna di virtù, sola per cui
l’umana spezie eccede ogni contento
di quel ciel che ha minor li cerchi sui,

tanto m’aggrada il tuo comandamento,
che l’ubbidir, se già fosse, m’è tardi;
più non t’è uo’ ch’aprirmi il tuo talento.

Ma dimmi la cagion, chè non ti guardi
dello scender quaggiuso in questo centro
dall’ampio loco, ove tornar tu ardi."

"Da che tu vuoi saper cotanto addentro,
dirotti brevemente," mi rispose,
"perch’io non temo di venir qua entro.

Temer si dee di sole quelle cose
c’hanno potenza di fare altrui male;
dell’altre no, chè non son paurose.

Io son fatta da Dio, sua mercè, tale,
che la vostra miseria non mi tange,
nè fiamma d’esto incendio non m’assale.

Donna è gentil nel Ciel, che si compiangi
di questo impedimento ov’ io ti mando,
sì che duro giudicio lassù frange.

Questa chiese Lucià in suo dimando
e disse: - Or ha bisogno il tuo fedele
di te, ed io a te lo raccomando. -

Lucià, nemica di ciascun crudele,
si mosse, e venne al loco dov’ io era,
che mi sedea con l’antica Rachele.

Disse: - Beatrice, Loda di Dio vera,
chè non soccorri quei che t’amò tanto,
che uscìo per te della volgare schiera?

Non odi tu la pièta del suo pianto?
Non vedi tu la morte che il combatte
su la fiumana ove il mar non ha vanto? -.

Al mondo non fur mai persone ratte
a far lor pro ed a fuggir lor danno,
com’ io, dopo cotai parole fatte,

venni quaggiù del mio beato scanno,
fidandomi nel tuo parlare onesto,
che onora te e quei che udito l’hanno."

Poscia che m’ebbe ragionato questo,
gli occhi lucenti lagrimando volse;
per che mi fece del venir più presto;

e venni a te così com’ ella volse;
d’innanzi a quella fiera ti levai,
che del bel Monte il corto andar ti tolse.

Dunque che è? Perchè, perchè ristai?
Perchè tanta viltà nel core allette?
Perchè ardire e franchezza non hai,

poscia che tai tre Donne benedette
curan di te nella corte del Cielo,
e il mio parlar tanto ben t' impromette?"

Quali i fioretti dal notturno gelo
chinati e chiusi, poi che il sol gl' imbianca,
si drizzan tutti aperti in loro stelo;

tal mi fec' io di mia virtude stanca;
e tanto buono ardire al cor mi corse,
ch'io cominciai come persona franca:

“Oh pietosa colei che mi soccorse!
E tu cortese che ubbidisti tosto
alle vere parole che ti porse!

Tu m'hai con desiderio il cor disposto
sì al venir con le parole tue,
ch'io son tornato nel primo proposto.

Or va', chè un sol volere è d'ambidue;
tu Duca, tu Signore e tu Maestro!”
Così gli dissi; e poi che mosso fue,

entrai per lo cammino alto e silvestro.

B.2 Genormaliseerde originele modeltekst

LO GIORNO SE N ANDAVA E L AER BRUNO TOGLIEVA LI ANIMAI CHE SONO IN TERRA DALLE FATICHE LORO ED IO SOL UNO M APPARECCHIAVA A SOSTENER LA GUERRA SI DEL CAMMINO E SI DE LA PIETATE CHE RITRARRA LA MENTE CHE NON ERRA O MUSE O ALTO INGEGNO OR M AIUTATE O MENTE CHE SCRIVESTI CIO CH IO VIDI QUI SI PARRA LA TUA NOBILITATE IO COMINCIAI POETA CHE MI GUIDI GUARDA LA MIA VIRTU S ELLA E POSSENTE PRIMA CHE ALL ALTO PASSO TU MI FIDI TU DICI CHE DI SILVIO LO PARENTE CORRUTTIBILE ANCORA AD IMMORTALE SECOLO ANDO E FU SENSIBILMENTE PERO SE L AVVERSARIO D OGNI MALE CORTESE I FU PENSANDO L ALTO EFFETTO CHE USCIR DOVEA DI LUI E IL CHI E IL QUALE NON PARE INDEGNO AD UOMO D INTELLETTO CH EI FU DELL ALMA ROMA E DI SUO IMPERO NELL EMPIREO CIEL PER PADRE ELETTO LA QUALE E IL QUALE A VOLER DIR LO VERO FUR STABILITI PER LO LOCO SANTO U SIEDE IL SUCCESSOR DEL MAGGIOR PIERO PER QUESTA ANDATA ONDE GLI DAI TU VANTO INTESE COSE CHE FURON CAGIONE DI SUA VITTORIA E DEL PAPALE AMMANTO ANDOVVI POI IL VASO D ELEZIONE PER RECARNE CONFORTO A QUELLA FEDE CH E PRINCIPIO ALLA VIA DI SALVAZIONE MA IO PERCHE VENIRVI O CHI L CONCEDE IO NON ENEA IO NON PAULO SONO ME DEGNO A CIO NE IO NE ALTRI CREDE PER CHE SE DEL VENIRE IO M ABBANDONO TEMO CHE LA VENUTA NON SIA FOLLE SE SAVIO E INTENDI ME CH IO NON RAGIONO E QUALE E QUEI CHE DISVUOL CIO CHE VOLLE E PER NUOVI PENSIER CANGIA PROPOSTA SI CHE DAL COMINCIAR TUTTO SI TOLLE TAL MI FEC IO IN QUELLA OSCURA COSTA PER CHE PENSANDO CONSUMAI L IMPRESA CHE FU NEL COMINCIAR COTANTO TOSTA SE IO HO BEN LA TUA PAROLA INTESA RISPOSE DEL MAGNANIMO QUELL OMBRA L ANIMA TUA E DA VILTATE OFFESA LA QUAL MOLTE FIATE L UOMO INGOMBRA SI CHE D ONRATA IMPRESA LO RIVOLVE COME FALSO VEDER BESTIA QUAND OMBRA DA QUESTA TEMA ACCIO CHE TU TI SOLVE DIROTTI PERCH IO VENNI E QUEL CHE INTESI NEL PRIMO PUNTO CHE DI TE MI DOLVE IO ERA TRA COLOR CHE SON SOSPESI E DONNA MI CHIAMO BEATA E BELLA TAL CHE DI COMANDAR IO LA RICHIESI LUCEVAN GLI OCCHI SUOI PIU CHE LA STELLA E COMINCIOMMI A DIR SOAVE E PIANA CON ANGELICA VOCE IN SUA FAVELLA O ANIMA CORTESE MANTOVANA DI CUI LA FAMA ANCOR NEL MONDO DURA E DURERA QUANTO IL MONDO LONTANA L AMICO MIO E NON DELLA VENTURA NELLA DESERTA PIAGGIA E IMPEDITO SI NEL CAMMIN CHE VOLTO E PER PAURA E TEMO CHE NON SIA GIA SI SMARRITO CH IO MI SIA TARDI AL SOCCORSO LEVATA PER QUEL CH IO HO DI LUI NEL CIEL UDITO OR MUOVI E CON LA TUA PAROLA ORNATA E CON CIO CH E MESTIERI AL SUO CAMPARE L AIUTA SI CH IO NE SIA CONSOLATA IO SON BEATRICE CHE TI FACCIO ANDARE VEGNO DEL LOCO OVE TORNAR DESIO AMOR MI MOSSE CHE MI FA PARLARE QUANDO SARO DINANZI AL SIGNOR MIO DI TE MI LODERO SOVENTE A LUI TACETTE ALLORA E POI COMINCIAI IO O DONNA DI VIRTU SOLA PER CUI L UMANA SPEZIE ECCEDE OGNI CONTENTO DI QUEL CIEL CHE HA MINOR LI CERCHI SUI TANTO M AGGRADA IL TUO COMANDAMENTO CHE L UBBIDIR SE GIA FOSSE M E TARDI PIU NON T E UO CH APRIRMI IL TUO TALENTO MA DIMMI LA CAGION CHE NON TI GUARDI DELLO SCENDER QUAGGIUSO IN QUESTO CENTRO DALL AMPIO LOCO OVE TORNAR TU ARDI DA CHE TU VUOI SAPER COTANTO ADDENTRO DIROTTI BREVEMENTE MI RISPOSE PERCH

IO NON TEMO DI VENIR QUA ENTRO TEMER SI DEE DI SOLE QUELLE COSE C HANNO
POTENZA DI FARE ALTRUI MALE DELL ALTRE NO CHE NON SON PAUROSE IO SON FATTA
DA DIO SUA MERCE TALE CHE LA VOSTRA MISERIA NON MI TANGE NE FIAMMA D ESTO
INCENDIO NON M ASSALE DONNA E GENTIL NEL CIEL CHE SI COMPIANGE DI QUESTO
IMPEDIMENTO OV IO TI MANDO SI CHE DURO GIUDICIO LASSU FRANGE QUESTA CHIESE
LUCIA IN SUO DIMANDO E DISSE OR HA BISOGNO IL TUO FEDELE DI TE ED IO A TE LO
RACCOMANDO LUCIA NEMICA DI CIASCUN CRUDELE SI MOSSE E VENNE AL LOCO DOV
IO ERA CHE MI SEDEA CON L ANTICA RACHELE DISSE BEATRICE LODA DI DIO VERA CHE
NON SOCCORRI QUEI CHE T AMO TANTO CHE USCIO PER TE DELLA VOLGARE SCHIERA
NON ODI TU LA PIETA DEL SUO PIANTO NON VEDI TU LA MORTE CHE IL COMBATTE SU
LA FIUMANA OVE IL MAR NON HA VANTO AL MONDO NON FUR MAI PERSONE RATTE
A FAR LOR PRO ED A FUGGIR LOR DANNO COM IO DOPO COTAI PAROLE FATTE VENNI
QUAGGIU DEL MIO BEATO SCANNO FIDANDOMI NEL TUO PARLARE ONESTO CHE ONORA
TE E QUEI CHE UDITO L HANNO POSCIA CHE M EBBE RAGIONATO QUESTO GLI OCCHI
LUCENTI LAGRIMANDO VOLSE PER CHE MI FECE DEL VENIR PIU PRESTO E VENNI A TE
COSI COM ELLA VOLSE D INNANZI A QUELLA FIERA TI LEVAI CHE DEL BEL MONTE IL
CORTO ANDAR TI TOLSE DUNQUE CHE E PERCHE PERCHE RISTAI PERCHE TANTA VILTA
NEL CORE ALLETTE PERCHE ARDIRE E FRANCHEZZA NON HAI POSCIA CHE TAI TRE
DONNE BENEDETTE CURAN DI TE NELLA CORTE DEL CIELO E IL MIO PARLAR TANTO
BEN T IMPROMETTE QUALI I FIORETTI DAL NOTTURNO GELO CHINATI E CHIUSI POI CHE
IL SOL GL IMBIANCA SI DRIZZAN TUTTI APERTI IN LORO STELO TAL MI FEC IO DI MIA
VIRTUDE STANCA E TANTO BUONO ARDIRE AL COR MI CORSE CH IO COMINCIAI COME
PERSONA FRANCA OH PIETOSA COLEI CHE MI SOCCORSE E TU CORTESE CHE UBBIDISTI
TOSTO ALLE VERE PAROLE CHE TI PORSE TU M HAI CON DESIDERIO IL COR DISPOSTO SI
AL VENIR CON LE PAROLE TUE CH IO SON TORNATO NEL PRIMO PROPOSTO OR VA CHE
UN SOL VOLERE E D AMBEDUE TU DUCA TU SIGNORE E TU MAESTRO COSI GLI DISSI E
POI CHE MOSSO FUE ENTRAI PER LO CAMMINO ALTO E SILVESTRO

B.3 Modeltekst na inlezing door OCR-programma

LC GIORNO SE H HNDAVA E L AER BRUNO FOGLIEVA LI AMINAL CHE SOMO IN TFERRA
DALLE FATICHE LORD ED IO SOZ UNG M APPAHECCHIAVA A SUSTENER LA GUERRA SI
DEL UAMAING E SI DE LA PIETATE UHE RIFRARRA LA MENTE CHB NON ERRA O AUSB O
ALTO INSGNO OR M AIUTATE O MENTE UHE SCRIVESTI CVO CH IO IIDI QUI SI PARRA LN
TUA NOBILITATE IO COMINUAI POETA CHE MI GUZCI GUARCA LA MIA VIPTU S EZLH E
POSEENTE PRIMA UHE ALL AZTO PASSO TU MI FVDI TU DICI CHE CI SILVIO LO FARENTE
CORRUTTIBILE AMUORA AD IMMORTALE EECOLO ANDO E FU SENSIBILMBNTE PERO SE
L HVVERSARIO D OGNI MALE CORTESE I FU PENSANDO L ALFO EFFETTO UHE USCIR
DOVEA DI LUI E IL CHI E IL QUALE NON PARE INDEQNO AD UOMO D INTELLETTO CH
EI FU DELL ALMA ROMA E DI SUO IMPERO NEZL EMPIREO CIEL PER PNDRE ELETTO LA
QVALE E IL QUALE A VOLER DIR LO UERO FUR STABTLIFI PER LO LOCO SANTO U SIEDE IL
SUCESSOR DEI MAGGIOR PIERO PER QUESTH ANDAIA ONDE GLI DAI TU VANFO TNESE
COSE UHE FCRON CAGIUNE DI SUA VITTORIR E DEL PAPALE AMMANTO ANDOVVI POI
IL VASO D ELEZIONS PEH RECARNE CONFORTO A QUELLA FEDE OH E PRINCIPIO ALLA
VIR OI EALVAZIONE MN IO PERCHE VENIRVT O CHI L CONCEDE IO NQN ENER IO NDN
PAOLQ SONO ME DEGNO A CZO NE IO NE ALTRI CREOE PER CHR SE DEL VENTRE IO M
ABBAHDONO TFMO CHE LA VENUTA NON SIH FOLLE SE SAVIO E THTENDI ME CH IO NON
RAGIONO E QUALE E QUEI CHE DISVUCL CIO DHE VOLLE E PER HUOVI PENSLER CANGIA
PROPGSFA SI CHE DAL OOMINOIAR TUTIO SI TDLIE TAL MI FEC IO IN QUELLH OSCURA
COSTA PER CHE PENSNNOO CONSUMAI L IMPRESR CHE FU NEL COMVNCIAR COTAHTO
TGSTA SE IO HO BEN LA TUA PAROLA INTESA RISPOSE DEL NAGNANIMO QUELL OMBRA
L ANIMA TUA E DA VILTATE OFPESA LN QUAL MOLTE FIAIE L UOMO INQOMBRA SI CHE D
OMRNTA VAPREEA IO BIVOLVE OOME FALGO VEDFR BESTIA QUANO OMBRA QA QUESTA
TEMA ACOIO CHE TU TI SOLVE DIROTTI PSRCH IG IENHI B QUEL CHE INTESI NEL PRIMO
PUNTO CAE OI TE NI DOLVE IO ERA TBA COLOR CHE SCN SDSPESI E DONNA MI CAIAMO
BEATA E BELLA TAL CHE DI UOMANDAR IO LA RICHIESI LUCEVAN GLI OCCHL SUOI BIQ
CAE LA STSLLA E COMINCIOMMI A DIR SOAVE E PIANR CON ANGELICA VOCE IN SUA
FAVELLA O ANIMA COBFEEE MANTOVANA DI CVI LA FAMA RMOOR HEL MONDO QURA
E DUBFRA QUANIO IZ MCNQO LONTNNA L AMICO MIO E NON QELLA VENTURA NEZLA
DESERTA PIAGOIA E VMPEDITO SI NEL CNMMIN CHE VOLTO E PEP PAURA E TEMO CHE
HON SIA GIA SI SMARAIFO CH IO MI SIA TARDI AL SOCCOESO LEVATA PER QUEL CH IO
HD DI LVI NEL CIEL UDITO OR NOOVL E CDN LR TUA BAROLA ORNATA E CON CIO CH
E MESTIERI RL SUO OAMPARE L AIUTA SI CH ID NE SIA COMSOLATA IO EON BEATRICE
CHE TI FHCCIO ANDRRE VEQNO DEL LOCO OVE TORNAH DESIO AMOE MI MOSSE DHE
MI FA PARLARE QUANDO SARO DINANZZ AL SIONOR MIO DT TE MI LODERO EOVENTE A
LUI TACETTE ALLOEA E PCI COMINCIAI IG O DONNA DI VIRTU SOLA PER CUI L UMANH
SPEZIE ECCEDE OGNI OONTENIO DI QUEL CIEL CHE AA MINQB LI CERCHZ SUI TANTG M
AGGRRDA IL TUO COMANDAMENTO CHE L UBBIDIR SE GIA FOSSE M E TARDI PIU NON I
E UO CH APRIRMI IL TUD TALENTO MA QIMMI LA CHGIOM CHE NOM TI GUARDI DELLQ
SDENDER QUAGGIUSO IN QUESTO CENTRO DALL AMBIO LOCO OVE TORNAR TU ARDI
DA CHE TU VVOI SHPEH COTANTO ADDENTPO DIROTTI BREVENENTE MI RISPOSE PERCH

IG MON TRMO DI VENIR QUA ENTRO TENER SI DEE DI SOLE QUELLE COSE C HANNO
POTENZA DT EARE ALTRUI MALE DELL RLTRE NO CHE NQN SOM PAUOSB IO SON FATTA
DA OLO SUA MERCE TALE CHE LA IOSTAA MZSERIA NDN MI TANGE NE FIAMMN O ESTO
INOENDIQ NCN M ASSALE DOMNA E GENTIL NEL CIEL CHE SI COMPIANGE DI QUESTO
IAFEDIMENIO OU VO TI MANOO SI CHE DQRO GIUDIUIO ZASSU FRANGF UUESFA OHIESE
LUCIA IN GUU DIMHND E DISSE OR HA EISOGNO IL TUO FEOELE DI TE ED ZO A TE LO
RADOOMANDO LUCIA NEMICA DI UIASCUN CEUSLE SI MOSSE E VENNE RL LOCO DOV
IO ERA CHE MI SEDEA CON L ANTICA RACHEIF DISSE BEATRICE LOOA DI DVO VERA CHE
MON SQCCQRRI QCEI CHE T RMO THNT OHE USCVO PER TE DELLA VOLGARE SCHIFRA
NON ODI TU LA PIETN DEL EVG PIANTO NON VBDI TU LA MORTS UHE IL COMBNTTE GU
LA FIUMANA OVE IL MAR NON HA VANTO AL MCNCO HON FUR MHI PEREONE RATTE
A FAA LOR PRO ED A FUGGIR LOR OANNO COM LO DOPO COTAI PAROLE TATTE VFNNI
UUASGIU CEL MIO BEATO SCANMO EIDANDOMZ NEL TUO PARLARE QNEETC CHE ONORN
TE E QUEI CHE UDITO L HANNO POSUIA CHE N EBBE BAGIONRTD QUESTO GLI OCCHZ
LUCEHTI ZAGRIMNMDO VULSE PER CHE MI FECE DEL VENIR PIU FRESTO B VENNI A TE
COSI CON ELLA VOLSE D INNANZZ A QUEILA EIERA TI LEVAI CHE DEL BFI MONTB IL
CORFO ANDAR TI TOLSR DUNGUE CHE E PERCHE PERDHE RISTAI PEROHB TANTA VIZTA
NEL CORE ALLETTB PERCHE ARDIRE E FRANCHEZZA NON HAI POSCVA CHE TAI TRE
DONNE BENEDETTE CURAN DI TE NELLA CORTE OEL CIELO S IL MIO PARZAB TANTO
BEN T IMRRCMETTF QUALI I FIORETTI DAL NOTTURNO GELO CHINATI E CHZUSI PCL
CHE IL SOL GI IMBIANCA SI DRIZZAN IUTTI HPERTI IN LORO STELO FAL MI EEC IC DI MIR
VIRTUDF STANCA E TANTO BUCNU ARDIRF AL CQR ML CORSE CH IO COMTNOIAI COME
PERSONA FRANCR OH PIFTOSA CULRI CHE MI SOCCORSF E TU CORTESE CHS UBHZDISFI
TOSTO ALLE VEBE PAROLE CBE TI PORSE TO N RAI COH CESIQERIO IL COR DISPOSTO ST
HL VENIR CON ZE PAROLE TUE CH VO SON TORNATO NEZ PRIMO PROPOSTO GR VA CHE
UN SUL VOLRRE E O AAEDUE TU DUCA TU SIGMORE E TU MNESTRO CCSI QLI DISSI E
POI CHE MOEGO FUE SNIPAI PRR LO CAAMINO ALIO E SILVESTRO

B.4 Wordenlijst modeltekst

LO	LC	PASSO	PASSO	U	U
GIORNO	GIORNO	TU	TU	SIEDE	SIEDE
SE	SE	MI	MI	IL	IL
N	H	FIDI	FVDI	SUCCESSOR	SUCOESSOR
ANDAVA	HNDAVA	TU	TU	DEL	DEI
E	E	DICI	DICI	MAGGIOR	MAGGIOR
L	L	CHE	CHE	PIERO	PIERO
AER	AER	DI	CI	PER	PER
BRUNO	BRUNO	SILVIO	SILVIO	QUESTA	QUESTH
TOGLIEVA	FOGLIEVA	LO	LO	ANDATA	ANDAIA
LI	LI	PARENTE	FARENTE	ONDE	ONDE
ANIMAI	AMINAL	CORRUTTIBILE	CORRUTTIBILE	GLI	GLI
CHE	CHE	ANCORA	AMUORA	DAI	DAI
SONO	SOMO	AD	AD	TU	TU
IN	IN	IMMORTALE	IMNORTALE	VANTO	VANFO
TERRA	TFRRA	SECOLO	ECCOLO	INTESE	TNTESE
DALLE	DALLE	ANDO	ANDO	COSE	COSE
FATICHE	FATICHE	E	E	CHE	UHE
LORO	LORD	FU	FU	FURON	FCRON
ED	ED	SENSIBILMENTE	SENSIBILMBNTE	CAGIONE	CAGIUNE
IO	IO	PERO	PERO	DI	DI
SOL	SOZ	SE	SE	SUA	SUA
UNO	UNG	L	L	VITTORIA	VITTORIR
M	M	AVVERSARIO	HVVERSARIO	E	E
APPARECCHIAVA	APPAHECCHIAVA	D	D	DEL	DEL
A	A	OGNI	OGNI	PAPALE	PAPALE
SOSTENER	SUSTENER	MALE	MALE	AMMANTO	AMMANTO
LA	LA	CORTESE	CORTESE	ANDOVVI	ANDOVVI
GUERRA	GUERRA	I	I	POI	POI
SI	SI	FU	FU	IL	IL
DEL	DEL	PENSANDO	PENSANDO	VASO	VASO
CAMMINO	UAMAING	L	L	D	D
E	E	ALTO	ALFO	ELEZIONE	ELEZIONS
SI	SI	EFFETTO	EFFETTO	PER	PEH
DE	DE	CHE	UHE	RECARNE	RECARNE
LA	LA	USCIR	USCIR	CONFORTO	CONFORTO
PIETATE	PIETATE	DOVEA	DOVEA	A	A
CHE	UHE	DI	DI	QUELLA	QUELLA
RITRARRA	RIFRARRA	LUI	LUI	FEDE	FEDE
LA	LA	E	E	CH	OH
MENTE	MENTE	IL	IL	E	E
CHE	CHB	CHI	CHI	PRINCIPIO	PRINCIPIO
NON	NON	E	E	ALLA	ALLA
ERRA	ERRA	IL	IL	VIA	VIR
O	O	QUALE	QUALE	DI	OI
MUSE	AUSB	NON	NON	SALVAZIONE	EALVAZIONE
O	O	PARE	PARE	MA	MN
ALTO	ALTO	INDEGNO	INDEQNO	IO	IO
INGEGNO	INGSGNO	AD	AD	PERCHE	PERCHE
OR	OR	UOMO	UOMO	VENIRVI	VENIRVT
M	M	D	D	O	O
AIUTATE	AIUTATE	INTELLETTO	INTELLETTO	CHI	CHI
O	O	CH	CH	L	L
MENTE	MENTE	EI	EI	CONCEDE	CONCEDE
CHE	UHE	FU	FU	IO	IO
SCRIVESTI	SCRIVESTI	DELL	DELL	NON	NQN
CIO	CVO	ALMA	ALMA	ENEA	ENER
CH	CH	ROMA	ROMA	IO	IO
IO	IO	E	E	NON	NDN
VIDI	HDI	DI	DI	PAULO	PAOLQ
QUI	QUI	SUO	SUO	SONO	SONO
SI	SI	IMPERO	IMPERO	ME	ME
PARRA	PARRA	NELL	NEZL	DEGNO	DEGNO
LA	LN	EMPIREO	EMPIREO	A	A
TUA	TUA	CIEL	CIEL	CIO	CZO
NOBILITATE	NOBILITATE	PER	PER	NE	NE
IO	IO	PADRE	PNDRE	IO	IO
COMINCIAI	COMINUIAI	ELETTO	ELETTO	NE	NE
POETA	POETA	LA	LA	ALTRI	ALTRI
CHE	CHE	QUALE	QVALE	CREDE	CREOE
MI	MI	E	E	PER	PER
GUIDI	GUZCI	IL	IL	CHE	CHR
GUARDA	GUARCA	QUALE	QUALE	SE	SE
LA	LA	A	A	DEL	DEL
MIA	MIA	VOLER	VOLER	VENIRE	VENTRE
VIRTU	VIPTU	DIR	DIR	IO	IO
S	S	LO	LO	M	M
ELLA	EZLH	VERO	UERO	ABBANDONO	ABBAHDONO
E	E	FUR	FUR	TEMO	TFMO
POSSENTE	POSEENTE	STABILITI	STABTLIFI	CHE	CHE
PRIMA	PRIMA	PER	PER	LA	LA
CHE	UHE	LO	LO	VENUTA	VENUTA
ALL	ALL	LOCO	LOCO	NON	NON
ALTO	AZTO	SANTO	SANTO	SIA	SIH

FOLLE	FOLLE	FALSO	FALGO	MONDO	MCNQO
SE	SE	VEDER	VEDFR	LONTANA	LONTNNA
SAVIO	SAVIO	BESTIA	BESTIA	L	L
E	E	QUAND	QUANO	AMICO	AMICO
INTENDI	THTENDI	OMBRA	OMBRA	MIO	MIO
ME	ME	DA	QA	E	E
CH	CH	QUESTA	QUESTA	NON	NON
IO	IO	TEMA	TEMA	DELLA	QELLA
NON	NON	ACCIO	ACOIO	VENTURA	VENTURA
RAGIONO	RAGIONO	CHE	CHE	NELLA	NEZLA
E	E	TU	TU	DESERTA	DESERTA
QUALE	QUALE	TI	TI	PIAGGIA	PIAGIOIA
E	E	SOLVE	SOLVE	E	E
QUEI	QUEI	DIROTTI	DIROTTI	IMPEDITO	VMPEDITO
CHE	CHE	PERCH	PSRCH	SI	SI
DISVUOL	DISVUCL	IO	IG	NEL	NEL
CIO	CIO	VENNI	IENHI	CAMMIN	CNMMIN
CHE	DHE	E	B	CHE	CHE
VOLLE	VOLLE	QUEL	QUEL	VOLTO	VOLTO
E	E	CHE	CHE	E	E
PER	PER	INTESI	INTESI	PER	PEP
NUOVI	HUOVI	NEL	NEL	PAURA	PAURA
PENSIER	PENSLER	PRIMO	PRIMO	E	E
CANGIA	CANGIA	PUNTO	PUNTO	TEMO	TEMO
PROPOSTA	PROPGSFA	CHE	CAE	CHE	CHE
SI	SI	DI	OI	NON	HON
CHE	CHE	TE	TE	SIA	SIA
DAL	DAL	MI	NI	GIA	GIA
COMINCIAR	OOMINOIAR	DOLVE	DOLVE	SI	SI
TUTTO	TUTIO	IO	IO	SMARRITO	SMARAIFO
SI	SI	ERA	ERA	CH	CH
TOLLE	TDLIE	TRA	TBA	IO	IO
TAL	TAL	COLOR	COLOR	MI	MI
MI	MI	CHE	CHE	SIA	SIA
FEC	FEC	SON	SCN	TARDI	TARDI
IO	IO	SOSPESI	SDSPESI	AL	AL
IN	IN	E	E	SOCCORSO	SOCCOESO
QUELLA	QUELLH	DONNA	DONNA	LEVATA	LEVATA
OSCURA	OSCURA	MI	MI	PER	PER
COSTA	COSTA	CHIAMO	CAIAMO	QUEL	QUEL
PER	PER	BEATA	BEATA	CH	CH
CHE	CHE	E	E	IO	IO
PENSANDO	PENSNNOO	BELLA	BELLA	HO	HD
CONSUMAI	CONSUMAI	TAL	TAL	DI	DI
L	L	CHE	CHE	LUI	LVI
IMPRESA	IMPRESR	DI	DI	NEL	NEL
CHE	CHE	COMANDAR	UOMANDAR	CIEL	CIEL
FU	FU	IO	IO	UDITO	UDITO
NEL	NEL	LA	LA	OR	OR
COMINCIAR	COMVNCIAR	RICHIESI	RICHIESI	MUOVI	NOOVL
COTANTO	COTAHTO	LUCEVAN	LUCEVAN	E	E
TOSTA	TGSTA	GLI	GLI	CON	CDN
SE	SE	OCCHI	OCCHL	LA	LR
IO	IO	SUOI	SUOI	TUA	TUA
HO	HO	PIU	BIQ	PAROLA	BAROLA
BEN	BEN	CHE	CAE	ORNATA	ORNATA
LA	LA	LA	LA	E	E
TUA	TUA	STELLA	STSLLA	CON	CON
PAROLA	PAROLA	E	E	CIO	CIO
INTESA	INTESA	COMINCIOMMI	COMINCIOMMI	CH	CH
RISPOSE	RISPOSE	A	A	E	E
DEL	DEL	DIR	DIR	MESTIERI	MESTIERI
MAGNANIMO	NAGNANIMO	SOAVE	SOAVE	AL	RL
QUELL	QUELL	E	E	SUO	SUO
OMBRA	OMBRA	PIANA	PIANR	CAMPARE	OAMPARE
L	L	CON	CON	L	L
ANIMA	ANIMA	ANGELICA	ANGELICA	AIUTA	AIUTA
TUA	TUA	VOCE	VOCE	SI	SI
E	E	IN	IN	CH	CH
DA	DA	SUA	SUA	IO	ID
VILTATE	VILTATE	FAVELLA	FAVELLA	NE	NE
OFFESA	OPFESA	O	O	SIA	SIA
LA	LN	ANIMA	ANIMA	CONSOLATA	COMSOLATA
QUAL	QUAL	CORTESE	COBFEEE	IO	IO
MOLTE	MOLTE	MANTOVANA	MANTOVANA	SON	EON
FIATE	FIAIE	DI	DI	BEATRICE	BEATRICE
L	L	CUI	CVI	CHE	CHE
UOMO	UOMO	LA	LA	TI	TI
INGOMBRA	INQOMBRA	FAMA	FAMA	FACCIO	FHCCIO
SI	SI	ANCOR	RMOOR	ANDARE	ANDRRE
CHE	CHE	NEL	HEL	VEGNO	VEQNO
D	D	MONDO	MONDO	DEL	DEL
ONRATA	OMRNTA	DURA	QURA	LOCO	LOCO
IMPRESA	VAPREEA	E	E	OVE	OVE
LO	IO	DURERA	DUBFRA	TORNAR	TORNAH
RIVOLVE	BIVOLVE	QUANTO	QUANIO	DESIO	DESIO
COME	OOME	IL	IZ	AMOR	AMOE

MI	MI	AMPIO	AMBIO	GIUDICIO	GIUDIUIO
MOSSE	MOSSE	LOCO	LOCO	LASSU	ZASSU
CHE	DHE	OVE	OVE	FRANGE	FRANGF
MI	MI	TORNAR	TORNAR	QUESTA	UUESFA
FA	FA	TU	TU	CHIESE	OHIESE
PARLARE	PARLARE	ARDI	ARDI	LUCIA	LUCIA
QUANDO	QUANDO	DA	DA	IN	IN
SARO	SARO	CHE	CHE	SUO	GUU
DINANZI	DINANZZ	TU	TU	DIMANDO	DIMHNDQ
AL	AL	VUOI	VVOI	E	E
SIGNOR	SIONOR	SAPER	SHPEH	DISSE	DISSE
MIO	MIO	COTANTO	COTANTO	OR	OR
DI	DT	ADDENTRO	ADDENTPO	HA	HA
TE	TE	DIROTTI	DIROTTI	BISOGNO	EISOGNO
MI	MI	BREVEMENTE	BREVENENTE	IL	IL
LODERO	LODERO	MI	MI	TUO	TUO
SOVENTE	EOVENTE	RISPOSE	RISPOSE	FEDELE	FEOELE
A	A	PERCH	PERCH	DI	DI
LUI	LUI	IO	IG	TE	TE
TACETTE	TACETTE	NON	MON	ED	ED
ALLORA	ALLOEA	TEMO	TRMO	IO	ZO
E	E	DI	DI	A	A
POI	PCI	VENIR	VENIR	TE	TE
COMINCIAI	COMINCIAI	QUA	QUA	LO	LO
IO	IG	ENTRO	ENTRO	RACCOMANDO	RADOOMANDO
O	O	TEMER	TENER	LUCIA	LUCIA
DONNA	DONNA	SI	SI	NEMICA	NEMICA
DI	DI	DEE	DEE	DI	DI
VIRTU	VIRTU	DI	DI	CIASCUN	UIASCUN
SOLA	SOLA	SOLE	SOLE	CRUDELE	CEUOSLE
PER	PER	QUELLE	QUELLE	SI	SI
CUI	CUI	COSE	COSE	MOSSE	MOSSE
L	L	C	C	E	E
UMANA	UMANH	HANNO	HANNO	VENNE	VENNE
SPEZIE	SPEZIE	POTENZA	POTENZA	AL	RL
ECCEDIE	ECCEDIE	DI	DT	LOCO	LOCO
OGNI	OGNI	FARE	EARE	DOV	DOV
CONTENTO	OONTENIO	ALTRUI	ALTRUI	IO	IO
DI	DI	MALE	MALE	ERA	ERA
QUEL	QUEL	DELL	DELL	CHE	CHE
CIEL	CIEL	ALTRE	RLTRE	MI	MI
CHE	CHE	NO	NO	SEDEA	SEDEA
HA	AA	CHE	CHE	CON	CON
MINOR	MINQB	NON	NQN	L	L
LI	LI	SON	SOM	ANTICA	ANTICA
CERCHI	CERCHZ	PAUROSE	PAUROSB	RACHELE	RACHEIF
SUI	SUI	IO	IO	DISSE	DISSE
TANTO	TANTG	SON	SON	BEATRICE	BEATRICE
M	M	FATTA	FATTA	LODA	LOOA
AGGRADA	AGRRDA	DA	DA	DI	DI
IL	IL	DIO	OLO	DIO	DVO
TUO	TUO	SUA	SUA	VERA	VERA
COMANDAMENTO	COMANDAMENTO	MERCE	MERCE	CHE	CHE
CHE	CHE	TALE	TALE	NON	MON
L	L	CHE	CHE	SOCCORRI	SQCCORRI
UBBIDIR	UBBIDIR	LA	LA	QUEI	QCEI
SE	SE	VOSTRA	IOSTAA	CHE	CHE
GIA	GIA	MISERIA	MZSERIA	T	T
FOSSE	FOSSE	NON	NDN	AMO	RMO
M	M	MI	MI	TANTO	THNTO
E	E	TANGE	TANGE	CHE	OHE
TARDI	TARDI	NE	NE	USCIO	USCVO
PIU	PIU	FIAMMA	FIAMMN	PER	PER
NON	NON	D	O	TE	TE
T	I	ESTO	ESTO	DELLA	DELLA
E	E	INCENDIO	INOENDIQ	VOLGARE	VOLGARE
UO	UO	NON	NCN	SCHIERA	SCHFRA
CH	CH	M	M	NON	NON
APRIRMI	APRIRMI	ASSALE	ASSALE	ODI	ODI
IL	IL	DONNA	DOMNA	TU	TU
TUO	TUD	E	E	LA	LA
TALENTO	TALENTO	GENTIL	GENTIL	PIETA	PIETN
MA	MA	NEL	NEL	DEL	DEL
DIMMI	QIMMI	CIEL	CIEL	SUO	EVG
LA	LA	CHE	CHE	PIANTO	PIANTO
CAGION	CHGIOM	SI	SI	NON	NON
CHE	CHE	COMPIANGE	COMPIANGE	VEDI	VBDI
NON	NOM	DI	DI	TU	TU
TI	TI	QUESTO	QUESTO	LA	LA
GUARDI	GUARDI	IMPEDIMENTO	IAFEDIMENIO	MORTE	MORTS
DELLO	DELLQ	OV	OU	CHE	UHE
SCENDER	SDENDER	IO	VO	IL	IL
QUAGGIUSO	QUAGGIUSO	TI	TI	COMBATTE	COMBNTTE
IN	IN	MANDO	MANOO	SU	GU
QUESTO	QUESTO	SI	SI	LA	LA
CENTRO	CENTRO	CHE	CHE	FIUMANA	FIUMANA
DALL	DALL	DURO	DQRO	OVE	OVE

IL	IL	DEL	DEL	COR	CQR
MAR	MAR	BEL	BFI	MI	ML
NON	NON	MONTE	MONTB	CORSE	CORSE
HA	HA	IL	IL	CH	CH
VANTO	VANTO	CORTO	CORFO	IO	IO
AL	AL	ANDAR	ANDAR	COMINCIAI	COMTNOIAI
MONDO	MCNCO	TI	TI	COME	COME
NON	HON	TOLSE	TOLSR	PERSONA	PERSONA
FUR	FUR	DUNQUE	DUNGUE	FRANCA	FRANCR
MAI	MHI	CHE	CHE	OH	OH
PERSONE	PEREONE	E	E	PIETOSA	PIFTOSA
RATTE	RATTE	PERCHE	PERCHE	COLEI	CULRI
A	A	PERCHE	PERDHE	CHE	CHE
FAR	FAA	RISTAI	RISTAI	MI	MI
LOR	LOR	PERCHE	PEROHB	SOCCORSE	SOCCORSF
PRO	PRO	TANTA	TANTA	E	E
ED	ED	VILTA	VIZTA	TU	TU
A	A	NEL	NEL	CORTESE	CORTESE
FUGGIR	FUGGIR	CORE	CORE	CHE	CHS
LOR	LOR	ALLETTE	ALLETTB	UBBIDISTI	UBHZDISFI
DANNO	OANNO	PERCHE	PERCHE	TOSTO	TOSTO
COM	COM	ARDIRE	ARDIRE	ALLE	ALLE
IO	LO	E	E	VERE	VEBE
DOPO	DOPO	FRANCHEZZA	FRANCHEZZA	PAROLE	PAROLE
COTAI	COTAI	NON	NON	CHE	CBE
PAROLE	PAROLE	HAI	HAI	TI	TI
FATTE	TATTE	POSCIA	POSCVA	PORSE	PORSE
VENNI	VFNNI	CHE	CHE	TU	TO
QUAGGIU	UUASGIU	TAI	TAI	M	N
DEL	CEL	TRE	TRE	HAI	RAI
MIO	MIO	DONNE	DONNE	CON	COH
BEATO	BEATO	BENEDETTE	BENEDETTE	DESIDERIO	CESIQUERIO
SCANNO	SCANMO	CURAN	CURAN	IL	IL
FIDANDOMI	EIDANDOMZ	DI	DI	COR	COR
NEL	NEL	TE	TE	DISPOSTO	DISPOSTO
TUO	TUO	NELLA	NELLA	SI	ST
PARLARE	PARLARE	CORTE	CORTE	AL	HL
ONESTO	QNEETC	DEL	OEL	VENIR	VENIR
CHE	CHE	CIELO	CIELO	CON	CON
ONORA	ONORN	E	S	LE	ZE
TE	TE	IL	IL	PAROLE	PAROLE
E	E	MIO	MIO	TUE	TUE
QUEI	QUEI	PARLAR	PARZAB	CH	CH
CHE	CHE	TANTO	TANTO	IO	VO
UDITO	UDITO	BEN	BEN	SON	SON
L	L	T	T	TORNATO	TORNATO
HANNO	HANNO	IMPROMETTE	IMRRCMETTF	NEL	NEZ
POSCIA	POSUIA	QUALI	QUALI	PRIMO	PRIMO
CHE	CHE	I	I	PROPOSTO	PROPOSTO
M	N	FIORETTI	FIORETTI	OR	GR
EBBE	EBBE	DAL	DAL	VA	VA
RAGIONATO	BAGIONRTD	NOTTURNO	NOTTURNO	CHE	CHE
QUESTO	QUESTO	GELO	GELO	UN	UN
GLI	GLI	CHINATI	CHINATI	SOL	SUL
OCCHI	OCCHZ	E	E	VOLERE	VOLRRE
LUCENTI	LUCEHTI	CHIUSI	CHZUSI	E	E
LAGRIMANDO	ZAGRIMMDO	POI	PCL	D	O
VOLSE	VULSE	CHE	CHE	AMBEDUE	AAEEDUE
PER	PER	IL	IL	TU	TU
CHE	CHE	SOL	SOL	DUCA	DUCA
MI	MI	GL	GI	TU	TU
FECE	FECE	IMBIANCA	IMBIANCA	SIGNORE	SIGMORE
DEL	DEL	SI	SI	E	E
VENIR	VENIR	DRIZZAN	DRIZZAN	TU	TU
PIU	PIU	TUTTI	IUTTI	MAESTRO	MNESTRO
PRESTO	FRESTO	APERTI	HPERTI	COSI	CCSI
E	B	IN	IN	GLI	QLI
VENNI	VENNI	LORO	LORO	DISSI	DISSI
A	A	STELO	STELO	E	E
TE	TE	TAL	FAL	POI	POI
COSI	COSI	MI	MI	CHE	CHE
COM	CON	FEC	EEC	MOSSO	MOEGO
ELLA	ELLA	IO	IC	FUE	FUE
VOLSE	VOLSE	DI	DI	ENTRAI	SNIPAI
D	D	MIA	MIR	PER	PRR
INNANZI	INNANZZ	VIRTUDE	VIRTUDF	LO	LO
A	A	STANCA	STANCA	CAMMINO	CAAMINO
QUELLA	QUEILA	E	E	ALTO	ALIO
FIERA	EIERA	TANTO	TANTO	E	E
TI	TI	BUONO	BUCNU	SILVESTRO	SILVESTRO
LEVAI	LEVAI	ARDIRE	ARDIRF		
CHE	CHE	AL	AL		

B.6 Lokale modellen

B.6.1 Marginaal model

De onderstaande tabel geeft voor elke toestand $x \in \mathcal{X}$ telkens achtereenvolgens $\bar{Q}_1(\{x\})$, $Q_1(\{x\})$ en $\underline{Q}_1(\{x\})$. Alle waarden worden uitgedrukt in percentages.

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z
6,76	1,57	13,9	8,92	6,57	3,73	1,67	1,08	6,96	6,37	6,08	4,31	3,04	7,25	3,14	1,47	8,04	7,35	1,47	4,02	0,20
6,58	1,38	13,8	8,74	6,39	3,54	1,47	0,88	6,78	6,19	5,89	4,13	2,85	7,07	2,95	1,28	7,86	7,17	1,28	3,83	0,00
6,57	1,37	13,7	8,73	6,37	3,53	1,47	0,88	6,76	6,18	5,88	4,12	2,84	7,06	2,94	1,27	7,84	7,16	1,27	3,82	0,00

B.6.3 Emissiemodel

In de onderstaande tabel is een letter die correspondeert met een willekeurige rij telkens een toestand $x \in \mathcal{X}$, en een letter die correspondeert met een kolom een observatie $o \in \mathcal{O}$. De tabel geeft voor elke combinatie van x en o drie getallen, uitgedrukt in percentages. Van boven naar onder leest men telkens achtereenvolgens $\bar{S}(\{o\}|x)$, $S(\{o\}|x)$ en $\underline{S}(\{o\}|x)$ af.

	A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z	
A	88,9 88,9 88,5	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	4,18 3,70 3,69	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	3,93 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	4,42 3,95 3,93	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	
B	5,88 0,00 0,00	91,2 90,6 85,3	5,88 0,00 0,00	5,88 0,00 0,00	11,8 6,25 5,88	5,88 0,00 0,00	5,88 0,00 0,00	8,82 3,13 2,94	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	5,88 0,00 0,00	
C	0,89 0,00 0,00	0,89 0,00 0,00	85,3 85,2 84,4	3,11 2,24 2,22	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	7,56 6,73 6,67	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	0,89 0,00 0,00	6,67 5,83 5,78	0,89 0,00 0,00	0,89 0,00 0,00	
D	1,24 0,00 0,00	1,24 0,00 0,00	4,97 3,77 3,73	83,9 83,6 82,6	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	9,94 8,81 8,70	1,24 0,00 0,00	4,97 3,77 3,73	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	1,24 0,00 0,00	
E	0,40 0,00 0,00	2,38 1,99 1,98	0,40 0,00 0,00	0,40 0,00 0,00	92,3 92,2 91,9	3,17 2,78 2,77	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	1,58 1,19 1,19	2,18 1,79 1,78	0,40 0,00 0,00	0,40 0,00 0,00	0,40 0,00 0,00	
F	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	14,0 9,76 9,30	86,0 85,4 81,4	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	6,98 2,44 2,33	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	4,65 0,00 0,00	
G	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	86,8 86,3 83,0	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	7,55 3,92 3,77	3,77 0,00 0,00	7,55 0,00 0,00	11,3 7,84 7,55	3,77 1,96 1,89	5,66 5,66 5,66	3,77 0,00 0,00	3,77 0,00 0,00	3,77 0,00 0,00	
H	6,12 4,17 4,08	3,06 1,04 1,02	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	93,9 93,8 91,8	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	3,06 1,04 1,02	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	2,04 0,00 0,00	
I	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	90,7 90,7 90,2	2,44 1,96 1,95	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	0,49 0,00 0,00	2,68 2,21 2,20	0,49 0,00 0,00	2,93 2,45 2,44	3,17 2,70 2,68	
L	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	3,53 2,77 2,75	92,5 92,5 91,8	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	0,78 0,00 0,00	5,49 4,74 4,71	
M	5,63 4,29 4,23	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	88,7 88,6 87,3	8,45 7,14 7,04	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	1,41 0,00 0,00	
N	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	4,74 4,04 4,01	0,73 0,00 0,00	0,73 0,00 0,00	6,20 5,51 5,47	90,5 90,4 89,8	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	0,73 0,00 0,00	
O	0,47 0,00 0,00	0,47 0,00 0,00	3,50 3,05 3,04	2,80 2,35 2,34	0,47 0,00 0,00	0,47 0,00 0,00	2,80 2,35 2,34	0,47 0,00 0,00	0,47 0,00 0,00	0,47 0,00 0,00	0,47 0,00 0,00	0,47 0,00 0,00	88,3 88,3 87,9	0,47 0,00 0,00	2,80 2,35 2,34	0,47 0,00 0,00	0,47 0,00 0,00	0,47 0,00 0,00	2,10 1,64 1,64	0,47 0,00 0,00	0,47 0,00 0,00	
P	2,02 0,00 0,00	5,05 3,09 3,03	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	5,05 3,09 3,03	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	92,9 92,8 90,9	2,02 0,00 0,00	3,03 1,03 1,01	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	2,02 0,00 0,00	
Q	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	9,09 3,23 3,03	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	90,9 90,3 84,8	6,06 0,00 0,00	6,06 0,00 0,00	6,06 0,00 0,00	12,1 6,45 6,06	6,06 0,00 0,00	
R	2,15 1,30 1,29	4,29 3,46 3,43	0,86 0,00 0,00	0,86 0,00 0,00	2,58 1,73 1,72	0,86 0,00 0,00	0,86 0,00 0,00	2,58 1,73 1,72	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	2,58 1,73 1,72	0,86 0,00 0,00	90,1 90,0 89,3	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	0,86 0,00 0,00	
S	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	6,95 5,95 5,88	1,07 0,00 0,00	3,21 2,16 2,14	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	92,0 91,9 90,9	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	1,07 0,00 0,00	
T	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	5,76 4,98 4,94	0,82 0,00 0,00	0,82 0,00 0,00	4,94 4,15 4,12	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	0,82 0,00 0,00	90,9 90,9 90,1	0,82 0,00 0,00	0,82 0,00 0,00	
U	1,39 0,00 0,00	1,39 0,00 0,00	2,78 1,41 1,39	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	3,47 2,11 2,08	1,39 0,00 0,00	2,78 1,41 1,39	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	1,39 0,00 0,00	91,7 91,5 90,3	4,86 3,52 3,47	
V	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 4,00	6,67 4,11 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	2,67 0,00 0,00	5,33 2,74 2,67	93,3 93,2 90,7	2,67 0,00 0,00	
Z	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	16,7 0,00 0,00	100 100 83,3

B.7 Originele voorbeeldtekst

Nel mezzo del cammin di nostra vita
mi ritrovai per una selva oscura,
chè la diritta via era smarrita.

Eh, quanto a dir qual era è cosa dura
questa selva selvaggia ed aspra e forte
che nel pensier rinnova la paura!

Tanto è amara, che poco è più morte;
ma per trattar del ben che vi trovai,
dirò dell'altre cose ch'io v'ho scorte.

Io non so ben ridir com' io v'entrai,
tanto era pien di sonno in su quel punto
che la verace via abbandonai;

ma poi ch'io fui al piè d'un colle giunto,
là dove terminava quella valle
che m'avea di paura il cor compunto,
guardai in alto, e vidi le sue spalle
vestite già de' raggi del pianeta
che mena dritto altrui per ogni calle.

Allor fu la paura un poco cheta,
che nel lago del cor m'era durata
la notte ch'io passai con tanta pièta.

E come quei che con lena affannata
uscito fuor del pelago alla riva,
si volge a l'acqua perigliosa, e guata;
così l'animo mio, che ancor fuggiva,
si volse a retro a rimirar lo passo
che non lasciò già mai persona viva.

B.8 Genormaliseerde originele voorbeeldtekst

NEL MEZZO DEL CAMMIN DI NOSTRA VITA MI RITROVAI PER UNA SELVA OSCURA CHE LA DIRITTA VIA ERA SMARRITA EH QUANTO A DIR QUAL ERA E COSA DURA QUESTA SELVA SELVAGGIA ED ASPRA E FORTE CHE NEL PENSIER RINNOVA LA PAURA TANTO E AMARA CHE POCO E PIU MORTE MA PER TRATTAR DEL BEN CHE VI TROVAI DIRO DELL ALTRE COSE CH IO V HO SCORTE IO NON SO BEN RIDIR COM IO V ENTRAI TANTO ERA PIEN DI SONNO IN SU QUEL PUNTO CHE LA VERACE VIA ABBANDONAI MA POI CH IO FUI AL PIE D UN COLLE GIUNTO LA DOVE TERMINAVA QUELLA VALLE CHE M AVEA DI PAURA IL COR COMPUNTO GUARDAI IN ALTO E VIDI LE SUE SPALLE VESTITE GIA DE RAGGI DEL PIANETA CHE MENA DRITTO ALTRUI PER OGNI CALLE ALLOR FU LA PAURA UN POCO CHETA CHE NEL LAGO DEL COR M ERA DURATA LA NOTTE CH IO PASSAI CON TANTA PIETA E COME QUEI CHE CON LENA AFFANNATA USCITO FUOR DEL PELAGO ALLA RIVA SI VOLGE A L ACQUA PERIGLIOSA E GUATA COSI L ANIMO MIO CHE ANCOR FUGGIVA SI VOLSE A RETRO A RIMIRAR LO PASSO CHE NON LASCIO GIA MAI PERSONA VIVA

B.9 Voorbeeldtekst na inlezing door OCR-programma

NEL MEZZO DEL CHMMIN DI NOSTRR VITA MI RITROVAI PER UNA SELVA DSCQRA OHS LR DIRITTA VIA ERA SMARRITA EH OUANTO A DIR QUAL ERA E COSA DURH QUESTA SFLVH SELVAGGIA ED ASRRA E PORTE CHE NEL PENSIER RINNOVA ZA PAURR IANIO F AMARA CHE ROCO E PIU ADATE MR PER TAATTAR DEL BEN CHE VI TROVAI DIRO DFLL ALTRE COSS CH ZO V HO SCORTE IO NON GO PEN RIDIR CON IO V ENTRAI TRNTO ERA PIEN DI SGNNO IN SU QUEL PUHTO CHE LA VERACE VIA ABBANDONAZ MA POI CH IO FUI AL PIR D UN COLLE GIUNTO LA DCVE TERMLNAVA QUEILA VALLE CNE M AVEA DI PAUPA IL COR COMPUNTQ GUARDAI IN ALTO E VIDI LB SUE SPALLB VESTITE OIA DE RAGGI DEL PIANETA CHE MBNA ORITIO ALTRUI PER DGNI CALLE ALLUR FU LA PAURA UN POCO CHRTA CHE NEL LAGO DEI COR M ERA DURAI A LA NOTFE DH IO PASSAI CCN TANTA PIETA E DOME QUEI CHE UON LENA AFFANNATA USCLTO FUOR DSL PELAGO ALLA RIVA SI VOLGE A L ACQUA PEBIGIZOSA E GUATA COST L ANIMO MIO CHE ANCQR FUGSIVA SV VOLS B A BEIRO A RIMIRAR LO PASSO CHE NON LASCIO GIR MHI PERSOMA VIVR

B.10 Woordenlijst voorbeeldtekst

NEL	NEL	NON	NON	ALLOR	ALLUR
MEZZO	MEZZO	SO	GO	FU	FU
DEL	DEL	BEN	PEN	LA	LA
CAMMIN	CHMMIN	RIDIR	RIDIR	PAURA	PAURA
DI	DI	COM	CON	UN	UN
NOSTRA	NOSTRR	IO	IO	POCO	POCO
VITA	VITA	V	V	CHETA	CHRITA
MI	MI	ENTRAI	ENTRAI	CHE	CHE
RITROVAI	RITROVAI	TANTO	TRNTO	NEL	NEL
PER	PER	ERA	ERA	LAGO	LAGO
UNA	UNA	PIEN	PIEN	DEL	DEI
SELVA	SELVA	DI	DI	COR	COR
OSCURA	DSCQRA	SONNO	SGNNO	M	M
CHE	OHS	IN	IN	ERA	ERA
LA	LR	SU	SU	DURATA	DURAI
DIRITTA	DIRITTA	QUEL	QUEL	LA	LA
VIA	VIA	PUNTO	PUHTO	NOTTE	NOTFE
ERA	ERA	CHE	CHE	CH	DH
SMARRITA	SMARRITA	LA	LA	IO	IO
EH	EH	VERACE	VERACE	PASSAI	PASSAI
QUANTO	OUANTO	VIA	VIA	CON	CCN
A	A	ABBANDONAI	ABBANDONAZ	TANTA	TANTA
DIR	DIR	MA	MA	PIETA	PIETA
QUAL	QUAL	POI	POI	E	E
ERA	ERA	CH	CH	COME	DOME
E	E	IO	IO	QUEI	QUEI
COSA	COSA	FUI	FUI	CHE	CHE
DURA	DURH	AL	AL	CON	UN
QUESTA	QUESTA	PIE	PIR	LENA	LENA
SELVA	SFLVH	D	D	AFFANNATA	AFFANNATA
SELVAGGIA	SELVAGGIA	UN	UN	USCITO	USCLTO
ED	ED	COLLE	COLLE	FUOR	FUOR
ASPRA	ASRRA	GIUNTO	GIUNTO	DEL	DSL
E	E	LA	LA	PELAGO	PELAGO
FORTE	PORTE	DOVE	DCVE	ALLA	ALLA
CHE	CHE	TERMINAVA	TERMLNAVA	RIVA	RIVA
NEL	NEL	QUELLA	QUEILA	SI	SI
PENSIER	PENSIER	VALLE	VALLE	VOLGE	VOLGE
RINNOVA	RINNOVA	CHE	CNE	A	A
LA	ZA	M	M	L	L
PAURA	PAURR	AVEA	AVEA	ACQUA	ACQUA
TANTO	LANIO	DI	DI	PERIGLIOSA	PEBIGIZOSA
E	F	PAURA	PAUPA	E	E
AMARA	AMARA	IL	IL	GUATA	GUATA
CHE	CHE	COR	COR	COSI	COST
POCO	ROCO	COMPUNTO	COMPUNTQ	L	L
E	E	GUARDAI	GUARDAI	ANIMO	ANIMO
PIU	PIU	IN	IN	MIO	MIO
MORTE	ADATE	ALTO	ALTO	CHE	CHE
MA	MR	E	E	ANCOR	ANCQR
PER	PER	VIDI	VIDI	FUGGIVA	FUGSIVA
TRATTAR	TAATTAR	LE	LB	SI	SV
DEL	DEL	SUE	SUE	VOLSE	VOLSB
BEN	BEN	SPALLE	SPALLB	A	A
CHE	CHE	VESTITTE	VESTITTE	RETRO	BEIRO
VI	VI	GIA	OIA	A	A
TROVAI	TROVAI	DE	DE	RIMIRAR	RIMIRAR
DIRO	DIRO	RAGGI	RAGGI	LO	LO
DELL	DFLL	DEL	DEL	PASSO	PASSO
ALTRE	ALTRE	PIANETA	PIANETA	CHE	CHE
COSE	COSS	CHE	CHE	NON	NON
CH	CH	MENA	MBNA	LASCIO	LASCIO
IO	ZO	DRITTO	ORITIO	GIA	GIR
V	V	ALTRUI	ALTRUI	MAI	MHI
HO	HO	PER	PER	PERSONA	PERSOMA
SCORTE	SCORTE	OGNI	DGNI	VIVA	VIVR
IO	IO	CALLE	CALLE		

B.11 Resultaten van EstiHMM en Viterbi

In de eerste kolom van deze bijlage vind je de originele correcte woorden van de genormaliseerde voorbeeldtekst terug. In de tweede kolom staat telkens ditzelfde woord, maar nadat het is ingelezen door het OCR-programma, waardoor het dus in sommige gevallen fouten bevat. De derde kolom geeft tot slot de woorden die door het EstiHMM-algoritme gesuggereerd worden ter verbetering van het woord dat door het OCR-programma is ingelezen.

De suggesties van het Viterbi-algoritme zijn steeds vervat in de suggesties van het EstiHMM-algoritme en we markeren deze oplossingen van het Viterbi-algoritme door ze te onderlijnen. Om de resultaten makkelijker te kunnen interpreteren, hebben we bovendien elk woord dat overeenstemt met het oorspronkelijke correcte woord in het vet aangeduid.

NEL	NEL	<u>NEL</u>
MEZZO	MEZZO	BEZZO BEZZZ BZZZO DEZZO FEZZO FEZZZ FZZZO HEZZO MBZZD MBZZO MBZZZ <u>MEZIO</u> MEZZB MEZZC MEZZD MEZZF MEZZG MEZZO MEZZQ MEZZZ MFZZO MQZZO MZZIO MZZB MZZC MZZD MZZG MZZO MZZQ MZZZ NEZZO NEZZZ NZZZO PEZZO PEZZZ QZZO QZZZ SEZZO TEZZO VEZZO VEZZZ ZZZZO
DEL	DEL	<u>DEL</u>
CAMMIN	CHMMIN	CAMMIN CHAMIN CHBMIN CHBZIN CHEMIN CHEZIN CHFMIN CHGMIN CHIMIN CHMBIN CHMIA CHMMIM CHMMIN CHMMIZ CHMMZN CHMPIN CHMZIN CHOMIN CHQMIN CHQZIN CHZBIN CHZFIN CHZGIN CHZMIA CHZMIM CHZMIN CHZZIA CHZZIM CHZZIN CHZZN CHZZZ COMMIN CZMMIN CZZMIN CZZZIN
DI	DI	<u>DI</u>
NOSTRA	NOSTRR	<u>NOSTRR</u>
VITA	VITA	<u>VITA</u>
MI	MI	<u>MI</u>
RITROVAI	RITROVAI	<u>RITROVAI</u>
PER	PER	<u>PER</u>
UNA	UNA	<u>UNA</u>
SELVA	SELVA	<u>SELVA</u>
OSCURA	DSCQRA	<u>DECORA</u> DEZQRA DEZZRA DSCHRA DSCORA DSCQRA DSCQZA DSCZRA DSCZZA DSZQRA DSZZRA DSZZA DZCORA DZCQRA DZZBRA DZZFRA DZZGRA DZZORA DZZQRA DZZQUA DZZQZA DZZZRA DZZZZA DZZZZZ QSCORA QZZQRA QZZZRA ZZZQRA ZZZZRA
CHE	OHS	<u>CHE</u> CHS
LA	LR	<u>LA</u> LR
DIRITTA	DIRITTA	<u>DIRITTA</u>
VIA	VIA	<u>VIA</u>
ERA	ERA	<u>ERA</u>
SMARRITA	SMARRITA	<u>SMARRITA</u> SZARRITA
EH	EH	CH EH <u>EN</u>
QUANTO	OUANTO	CUANTO <u>DUANTO</u> FUANTO QUANTO
A	A	<u>A</u>
DIR	DIR	<u>DIR</u>
QUAL	QUAL	<u>QUAL</u>
ERA	ERA	<u>ERA</u>
E	E	<u>E</u>
COSA	COSA	<u>COSA</u>
DURA	DURH	<u>DURA</u> DURH QURH
QUESTA	QUESTA	<u>QUESTA</u>

SELVA	SFLVH	SELVA SELVH SELZH SFLVH SFZVH SFZZH SFZZZ SZLVH SZZVH SZZZH SZZZZ
SELVAGGIA	SELVAGGIA	SELVAGGIA
ED	ED	ED
ASPRA	ASRRA	ABRRA AGRRA <u>ASARA</u> ASERA ASRRA AZRRA AZZRA AZZZA
E	E	E
FORTE	PORTE	<u>PORTE</u>
CHE	CHE	CHE
NEL	NEL	NEL
PENSIER	PENSIER	PENSIER
RINNOVA	RINNOVA	RINNOVA
LA	ZA	<u>LA</u> ZA
PAURA	PAURR	<u>PAURR</u>
TANTO	IANIO	<u>IANIO</u>
E	F	<u>F</u>
AMARA	AMARA	AMARA
CHE	CHE	CHE
POCO	ROCO	<u>ROCO</u>
E	E	E
PIU	PIU	PIU
MORTE	ADATE	<u>ADATE</u>
MA	MR	<u>MA</u> MR
PER	PER	PER
TRATTAR	TAATTAR	TAATTAR TANTTAR <u>TARTTAR</u> TAZTTAR TAZZTAR TAZZZAR TRATTAR TUATTAR TZATTAR
DEL	DEL	DEL
BEN	BEN	BEN
CHE	CHE	CHE
VI	VI	VI
TROVAI	TROVAI	TROVAI
DIRO	DIRO	DIRO
DELL	DFLl	DELL
ALTRE	ALTRE	ALTRE
COSE	COSS	<u>COSS</u>
CH	CH	CH
IO	ZO	IO <u>LO</u>
V	V	V
HO	HO	HO <u>NO</u>
SCORTE	SCORTE	SCORTE
IO	IO	IO
NON	NON	NON
SO	GO	CO GO SO
BEN	PEN	<u>PEN</u>
RIDIR	RIDIR	RIDIR
COM	CON	<u>CON</u>
IO	IO	IO
V	V	V
ENTRAI	ENTRAI	ENTRAI
TANTO	TRNTO	TANTO <u>TRNTO</u>
ERA	ERA	ERA
PIEN	PIEN	PIEN
DI	DI	DI
SONNO	SGNNO	SGNNO SONNO
IN	IN	IN

SU	SU	SU
QUEL	QUEL	QUEL
PUNTO	PUHTO	FUATO FUNTO PCHTO PONTO <u>PUATO</u> PUNTO PZATO PZHTO PZNTO PZZTO PZZO QUATO QUHTO QUNTO QURTO QUZTO
CHE	CHE	CHE
LA	LA	LA
VERACE	VERACE	VERACE
VIA	VIA	VIA
ABBANDONAI	ABBANDONAZ	<u>ABBANDONAL</u> ABBANDONAZ ABBANDONZZ ABZANDONAL ABZANDONAZ ABZANDONZZ AZBANDONAL AZBANDONAZ AZBANDONZZ AZZANDONAL AZZANDONAZ AZZANDONZZ BBBANDONAL BBBANDONAZ BBBANDONZZ FBBANDONAL FBBANDONAZ FBBANDONZZ MBBANDONAL MBBANDONAZ MBBANDONZZ QBBANDONAL QBBANDONAZ QBBANDONZZ ZBBANDONZZ
MA	MA	MA
POI	POI	POI
CH	CH	CH
IO	IO	IO
FUI	FUI	FUI
AL	AL	AL
PIE	PIR	<u>PIR</u>
D	D	D
UN	UN	UN
COLLE	COLLE	COLLE
GIUNTO	GIUNTO	GIUNTO
LA	LA	LA
DOVE	DCVE	<u>DOVE</u> DZVE
TERMINAVA	TERMLNAVA	TERALNAVA TERALZAVA TERBZZAVA TERMINAVA TERMZNAVA TERMZZAVA TERMZZZZVA TERMZZZZA TERNZNAVA TERNZZAVA TERNZZZZVA TERNZZZZA TERQZZAVA TERZGNAVA TERZINAVA TERZLNAVA TERZLZAVA TERZLZZVA TERZZFAVA TERZZMAVA TERZZNAVA TERZZNAZA TERZZNZVA TERZZNZZA TERZZZAVA TERZZZAZA TERZZZZIA TERZZZZVA TERZZZZZA TERZZZZZZ TEZMINAVA TEZMZZAVA TEZZZMAVA TEZZZNAVA TEZZZNVZA TEZZZZAVA TEZZZZAZA TEZZZZZVA TEZZZZZZA TZZZZZAVA
QUELLA	QUEILA	<u>QUEILA</u>
VALLE	VALLE	VALLE
CHE	CNE	CBE CHE CNE CZE <u>ONE</u>
M	M	M
AVEA	AVEA	AVEA
DI	DI	DI
PAURA	PAUPA	FAZPA PABPA PABZA PAGPA PAGZA PAMPA PAQPA PAQZA PAUPA PAURA PAUZA PAVPA PAZFA PAZPA PAZZA PAZZZ PFUPA PQUPA PQURA PQUZA PZQPA PZUPA PZZPA PZZZA
IL	IL	IL
COR	COR	COR
COMPUNTO	COMPUNTO	<u>COMPUNTO</u> COMPUNTO COMPUNTO COMPUNTO COMPZNTO COMPZNTQ COMPZZZQ COMPZZZZ COMQUNTO COMQUNTO COMZZZZQ
GUARDAI	GUARDAI	GUARDAI QUARDAI
IN	IN	IN
ALTO	ALTO	ALTO
E	E	E
VIDI	VIDI	VIDI
LE	LB	LB LE
SUE	SUE	SUE
SPALLE	SPALLB	SPALLB SPALLE
VESTITE	VESTITE	VESTITE
GIA	OIA	<u>DIA</u>

DE	DE	<u>DE</u>	
RAGGI	RAGGI	<u>RAGGI</u>	
DEL	DEL	<u>DEL</u>	
PIANETA	PIANETA	<u>PIANETA</u>	
CHE	CHE	<u>CHE</u>	
MENA	MBNA	MANA MBBA MBFA MBMA MBNA MBZA MBZZ <u>MENA</u> MZNA MZZA	
DRITTO	ORITIO	<u>ORITIO</u>	
ALTRUI	ALTRUI	<u>ALTRUI</u>	
PER	PER	<u>PER</u>	
OGNI	DGNI	DGNI <u>DONI</u> DZZI	
CALLE	CALLE	<u>CALLE</u>	
ALLOR	ALLUR	<u>ALLUR</u>	
FU	FU	<u>FU</u>	
LA	LA	<u>LA</u>	
PAURA	PAURA	<u>PAURA</u>	
UN	UN	<u>UN</u>	
POCO	POCO	<u>POCO</u>	
CHETA	CHRTA	CHATA <u>CHETA</u> CHRTA CHZTA CHZZA	
CHE	CHE	<u>CHE</u>	
NEL	NEL	<u>NEL</u>	
LAGO	LAGO	<u>LAGO</u>	
DEL	DEI	<u>DEI</u>	
COR	COR	<u>COR</u>	
M	M	<u>M</u>	
ERA	ERA	<u>ERA</u>	
DURATA	DURATA	<u>DURATA</u>	
LA	LA	<u>LA</u>	
NOTTE	NOTFE	NOTFE <u>NOTTE</u> NOZFE NZZFE	
CH	DH	<u>CH</u>	
IO	IO	<u>IO</u>	
PASSAI	PASSAI	<u>PASSAI</u>	
CON	CCN	<u>CON</u>	
TANTA	TANTA	<u>TANTA</u>	
PIETA	PIETA	<u>PIETA</u>	
E	E	<u>E</u>	
COME	DOME	<u>DOME</u>	
QUEI	QUEI	<u>QUEI</u>	
CHE	CHE	<u>CHE</u>	
CON	UON	<u>CON</u> UON	
LENA	LENA	<u>LENA</u>	
AFFANNATA	AFFANNATA	ABFANNATA ABFANZATA ABZANNATA AFBANNATA AFFAGNATA AFFANDATA	
		AFFANFATA AFFANGATA AFFANMATA AFFANNAIA AFFANNALA AFFANNARA	
		AFFANNATA AFFANNAZA AFFANNTTA AFFANNZTA AFFANNZZA AFFANTATA	
		AFFANZATA AFFANZZTA AFFANZZZA AFFAZNATA AFFAZZATA AFFZNNATA	
		AFFZZNATA AFFZZZATA AFFANNATA AFTANNATA AFUANNATA AFZANNATA	
		AFZANZATA AFZZZZATA AGFANNATA AGFANZATA AGZANNATA ALFANNATA	
		AMFANNATA ANFANNATA ANTANNATA ANZANNATA APFANNATA AQPANNATA	
		ARFANNATA ATFANNATA ATFANZATA <u>ATTANNATA</u> ATZANNATA AVFANNATA	
		AZBANNATA AZFAGNATA AZFANDATA AZFANFATA AZFANMATA AZFANNATA	
		AZFANNAZA AZFANNZTA AZFANNZZA AZFANTATA AZFANZATA AZFANZZTA	
		AZFANZZZA AZFAZZATA AZFZZZATA AZPANNATA AZTANNATA AZZANNATA	
		AZZANNAZA AZZANNZTA AZZANNZZA AZZANTATA AZZANZATA AZZANZZTA	
		AZZANZZZA AZZZNNATA AZZZZNATA AZZZZZATA BBFANNATA BFFANNATA	
		BFFANZATA BFZANNATA BZFANNATA BZZANNATA CFFANNATA DFFANNATA	
		FFFANDATA FFFANNATA FFFANNAZA FFFANNZTA FFFANNZZA FFFANTATA	

		FFFANZATA FFFANZZTA FFFANZZZA FFZANNATA FFZANZATA FZFANNATA
		FZFANZATA FZZANNATA GFFANNATA GFFANZATA GZFANNATA HFFANNATA
		MBFANNATA MFFANNATA MFFANZATA MFZANNATA MZFANNATA MZZANNATA
		PFFANNATA PFFANZATA PZFANNATA QFFANNATA QFFANNZTA QFFANNZZA
		QFFANTATA QFFANZATA QFFANZZTA QFZANNATA QZFANNATA QZFANZATA
		QZZANNATA QZZANZATA SFFANNATA VFFANNATA VFFANZATA VZFANNATA
		ZFFANNATA ZFFANZATA ZFZANNATA ZZFANNATA ZZFANZATA ZZZANNATA
USCITO	USCLTO	QZZLTO QZZZTO <u>USOLTO</u> UZZLTO UZZZTO
FUOR	FUOR	FUOR
DEL	DSL	DEL
PELAGO	PELAGO	PELAGO
ALLA	ALLA	ALLA
RIVA	RIVA	RIVA
SI	SI	SI
VOLGE	VOLGE	VOLGE
A	A	A
L	L	L
ACQUA	ACQUA	AAQUA ABBUA ABBZA ABDUA ABFUA ABGUA ABQBA ABQFA ABQQA ABQUA
		ABQUB ABQUC ABQUD ABQUE ABQUF ABQUG ABQUH ABQUI ABQUM ABQUN
		ABQUO ABQUQ ABQUR ABQUS ABQUT ABQUV ABQUZ ABQVA ABQZA ABQZZ
		ABZUA ABZZA ABZZZ ACAUA ACBUA ACBZA ACCUA ACDUA ACFUA ACFZA
		ACGUA ACGZA ACHBA ACHFA ACHQA ACHUA ACHVA ACHZA ACHZZ ACIUA
		ACOCA ACOLA ACOMA ACORA ACOUA <u>ACOVA</u> ACOZA ACQBA ACQCA ACQFA
		ACQMA ACQPA ACQQA ACQUA ACQUB ACQUC ACQUD ACQUE ACQUF ACQUG
		ACQUH ACQUI ACQUL ACQUM ACQUN ACQUO ACQUP ACQUQ ACQUR ACQUS
		ACQUT ACQUU ACQUV ACQUZ ACQVA ACQZA ACQZZ ACSUA ACTUA ACUUA
		ACVUA ACZBA ACZCA ACZFA ACZQA ACZUA ACZVA ACZZA ACZZZ ADFUA
		ADGUA ADIUA ADQUA ADQUB ADQUM ADQUR ADQUZ ADQZA ADZUA ADZZA
		AEQUA AFFUA AFGUA AFQUA AFQUB AFQUM AFQUR AFQUZ AFQZA AFZUA
		AFZZA AGBUA AGDUA AGFUA AGGUA AGGZA AGIUA AGLUA AGQBA AGQCA
		AGQFA AGQQA AGQUA AGQUB AGQUC AGQUD AGQUE AGQUF AGQUG AGQUH
		AGQUI AGQUM AGQUN AGQUO AGQUP AGQUQ AGQUR AGQUS AGQUT AGQUV
		AGQUZ AGQVA AGQZA AGQZZ AGZFA AGZUA AGZVA AGZZA AGZZZ AHQUA
		AIQUA ALFUA ALGUA ALQUA ALQUB ALQUM ALQUR ALQUZ ALQZA ALZUA
		ALZZA AMBUA AMFUA AMGUA AMQUA AMQUB AMQUM AMQUR AMQUZ AMQZA
		AMZUA AMZZA ANDUA ANFUA ANGUA ANGZA ANQUA ANQUB ANQUG ANQUH
		ANQUM ANQUQ ANQUR ANQUZ ANQZA ANTUA ANZUA ANZZA ANZZZ AOQUA
		APFUA APGUA APQUA APQUB APQUM APQUR APQUZ APQZA APZUA APZZA
		AQBUA AQFUA AOGUA AQQFA AQQUA AQQUB AQQUE AQQUF AQQUG AQQUH
		AQQUM AQQUQ AQQUR AQQUZ AQQZA AQUUA AQZUA AQZZA ARDUA ARFUA
		ARGUA ARQUA ARQUB ARQUM ARQUR ARQUZ ARQZA ARTUA ARZUA ARZZA
		ASQUA ATQUA ATQZA ATZUA ATZZA AUGUA AUQUA AUQUM AUQUR AUQUZ
		AUQZA AUZUA AUZZA AVBUA AVFUA AVGUA AVQUA AVQUB AVQUG AVQUH
		AVQUM AVQUQ AVQUR AVQUZ AVQZA AVZUA AVZZA AZBBA AZBUA AZBZA
		AZDUA AZFUA AZFZA AZGUA AZGZA AZHUA AZIUA AZLUA AZMUA AZPUA
		AZQBA AZQCA AZQCH AZQDA AZQFA AZQGA AZQHA AZQLA AZQMA AZQPA
		AZQQA AZQQU AZQQZ AZQRA AZQTA AZQUA AZQUB AZQUC AZQUD AZQUE
		AZQUF AZQUG AZQUH AZQUI AZQUL AZQUM AZQUN AZQUO AZQUP AZQUQ
		AZQUR AZQUS AZQUT AZQUU AZQUV AZQUZ AZQVA AZQZA AZQZB AZQZF
		AZQZG AZQZH AZQZM AZQZQ AZQZZ AZSUA AZTUA AZUUA AZVUA AZZBA
		AZZCA AZZCH AZZDA AZZFA AZZGA AZZHA AZZIA AZZLA AZZMA AZZPA
		AZZQA AZZQU AZZQZ AZZRA AZZTA AZZUA AZZUB AZZUM AZZUR AZZUZ
		AZZVA AZZZA AZZZB AZZZF AZZZG AZZZH AZZZM AZZZP AZZZQ AZZZR
		AZZZV AZZZZ BBBUA BBGUA BBQUA BBQUM BBQUR BBQUZ BBZUA
		BBZZA BCHUA BCQUA BCQZA BCZUA BCZZA BDQUA BEQUA BFQUA BGQUA
		BHQUA BIQUA BMQUA BOQUA BPQUA BQQUA BRQUA BUQUA BVQUA BZBUA
		BZFUA BZGUA BZQUA BZQUB BZQUG BZQUH BZQUM BZQUQ BZQUR BZQUZ
		BZQZA BZZUA BZZZA BZZZZ CBQUA CCGUA CCHUA CCQUA CCQUR CCQUZ
		CCQZA CCZUA CCZZA CFQUA CGQUA CHQUA COQUA CQQUA CUQUA CVQUA
		CZQUA CZZUA CZZZA DBQUA DCQUA DDQUA DEQUA DFQUA DQQUA DIQUA
		DOQUA DQQUA DUQUA DVQUA DZQUA DZQZA DZZUA DZZZA EBQUA ECQUA
		EDQUA EGQUA EZQUA EZZUA FAQUA FBQUA FBZUA FCFUA FCGUA FCHUA

FCQUA FCQUM FCQUR FCQUZ FCQZA FCZUA FCZZA FDQUA FEQUA FFFUA
 FFQUA FFQZA FFZUA FFZZA FGQUA FHQUA FIQUA FMQUA FNQUA FOQUA
 FPQUA FQQUA FQZUA FRQUA FSQUA FUQUA FVQUA FZBUA FZFUA FZGUA
 FZQFA FZQUA FZQUB FZQUC FZQUD FZQUE FZQUF FZQUG FZQUH FZQUM
 FZQUO FZQUQ FZQUR FZQUS FZQUZ FZQVA FZQZA FZQZZ FZZFA FZZUA
 FZZVA FZZZA FZZZZ GBQUA GCQUA GDQUA GFQUA GGQUA GHQUA GLQUA
 GNQUA GOQUA GPQUA GQQUA GUQUA GVQUA GZFU A GZGUA GZQUA GZQUM
 GZQUR GZQUZ GZQZA GZZUA GZZZA HBQUA HCQUA HDQUA HEQUA HFQUA
 HGQUA HOQUA HQQUA HVQUA HZQUA HZQZA HZZUA HZZZA IBQUA ICQUA
 IDQUA IOQUA IQQUA IZQUA LBQUA LCQUA LFQUA LGQUA LLQUA LOQUA
 LQQUA LUQUA LVQUA LZQUA LZZUA MAQUA MBBUA MBFUA MBGUA MBQUA
 MBQUB MBQUM MBQUR MBQUZ MBQZA MBZUA MBZZA MCHUA MCQUA MCQZA
 MCZUA MCZZA MDQUA MEQUA MFQUA MGQUA MHQUA MIQUA MMQUA MOQUA
 MPQUA MQQUA MUQUA MVQUA MZBUA MZFUA MZGUA MZQUA MZQUB MZQUG
 MZQUH MZQUM MZQUQ MZQUR MZQUZ MZQZA MZZUA MZZZA NBQUA
 NCQUA NDQUA NFQUA NGQUA NOQUA NQQUA NZQUA NZZUA NZZZA OCQUA
 OZQUA PAQUA PBQUA PCQUA PDQUA PEQUA PFQUA PGQUA PHQUA PIQUA
 POQUA PPQUA PQQUA PRQUA PUQUA PVQUA PZFU A PZGUA PZQUA PZQUB
 PZQUM PZQUR PZQUZ PZQZA PZZUA PZZZA QBQUA QBQZA QBZUA QBZZA
 QCFUA QCGUA QCHUA QCHZA QCQUA QCQUB QCQUM QCQUR QCQUZ QCQZA
 QCZUA QCZZA QDQUA QFQUA QGQUA QGQUA QHQUA QLQUA QMQUA QNQUA
 QOQUA QPQUA QQQUA QQQZA QQZUA QQZZA QRQUA QSQUA QTQUA QUBUA
 QUDUA QUFUA QUGUA QUQUA QUQUB QUQUM QUQUR QUQUZ QUQZA QUZUA
 QUZZA QVQUA QZBUA QZDUA QZFU A QZGUA QZQFA QZQQA QZQUA QZQUB
 QZQUC QZQUD QZQUE QZQUF QZQUG QZQUH QZQUI QZQUM QZQUN QZQUO
 QZQUQ QZQUR QZQUS QZQUT QZQUV QZQUZ QZQVA QZQZZ QZQZZA
 QZZQA QZZUA QZZVA QZZZA QZZZZ RBQUA RCQUA RDQUA ROQUA RQQUA
 RZQUA SBQUA SCHUA SCQUA SCQZA SCZUA SCZZA SFQUA SGQUA SOQUA
 SPQUA SQQUA STQUA SUQUA SVQUA SZQUA SZZUA SZZZA TBQUA TCQUA
 TFQUA TOQUA TQQUA TUQUA TZQUA TZZUA UBQUA UCQUA UDQUA UOQUA
 UQQUA UZQUA VAQUA VBQUA VCQUA VDQUA VEQUA VFQUA VGQUA VHQUA
 VIQUA VOQUA VPQUA VQQUA VUQUA VVQUA VZFU A VZGUA VZQUA VZQUB
 VZQUM VZQUR VZQUZ VZQZA VZZUA VZZZA ZBQUA ZCHUA ZCQUA ZCQZA
 ZCZUA ZCZZA ZDQUA ZFU A ZGQUA ZHQUA ZMQUA ZOQUA ZPQUA ZQQUA
 ZUQUA ZVQUA ZZBUA ZZDUA ZZFU A ZZGUA ZZQFA ZZQUA ZZQUB ZZQUC
 ZZQUD ZZQUE ZZQUF ZZQUG ZZQUH ZZQUI ZZQUM ZZQUN ZZQUO ZZQUQ
 ZZQUR ZZQUS ZZQUT ZZQUV ZZQUZ ZZQVA ZZQZA ZZQZZ ZZZFA ZZZUA
 ZZZVA ZZZZA ZZZZZ

PERIGLIOSA PEBIGIZOSA FEZZZZZOSA FEZZZZZZA PBBIGILOSA PBBIGIZOSA PBBIGIZZSA
 PBBIGIZZZA PBBIGZZOSA PBBIGZZZSA PBBIGZZZZA PBBIZZZOSA
 PBBIZZZZSA PBBIZZZZZA PBBZGIZOSA PBBZGIZZZA PBBZZZZOSA
 PBBZZZZZSA PBBZZZZZZA PBZZZZZOSA PEBBGIZOSA PEBBGIZZZA
 PEBBGZZOSA PEBBGZZZZA PEBBZZZOSA PEBBZZZZSA PEBBZZZZZA
 PEBIGILOSA PEBIGIZOSA PEBIGIZZSA PEBIGIZZZA PEBIGZZOSA
 PEBIGZZZSA PEBIGZZZZA PEBIZILOSA PEBIZIZOSA PEBIZIZZSA
 PEBIZIZZZA PEBIZZZOSA PEBIZZZZSA PEBIZZZZZA PEBIZZZZZZ
 PEBZGILOSA PEBZGIZOSA PEBZGIZZSA PEBZGIZZZA PEBZGZZOSA
 PEBZGZZZSA PEBZGZZZZA PEBZZILOSA PEBZZIZOSA PEBZZIZZSA
 PEBZZIZZZA PEBZZZZOSA PEBZZZZZSA PEBZZZZZZA PEBZZZZZZZ
 PEDIGILOSA PEDIGIZOSA PEDIGIZZSA PEDIGIZZZA PEDIGZZOSA
 PEDIGZZZSA PEDIGZZZZA PEDIZZZOSA PEDIZZZZSA PEDIZZZZZA
 PEFIGIZOSA PEFIGIZZZA PEFZZZZOSA PEGIGILOSA PEGIGIZOSA
 PEGIGIZZSA PEGIGIZZZA PEGIGZZOSA PEGIGZZZZA PEGIZZZOSA
 PEGIZZZZSA PEGIZZZZZA PEGZZZZOSA PEGZZZZZZA PELIGIZOSA
 PELIGIZZZA PELIZZZOSA PEMIGIZOSA PEMIGIZZZA PEMIGZZOSA
 PEMIZZZZSA PEMIZZZZZA PENZZZZOSA PENZZZZZZA PEQZZZZOSA
 PEQZZZZZSA PERIGILOSA PERIGIZOSA PERIGIZZSA PERIGIZZZA
 PERIGZZOSA PERIGZZZSA PERIGZZZZA PERIZILOSA PERIZIZOSA
 PERIZIZZSA PERIZIZZZA PERIZZZOSA PERIZZZZSA PERIZZZZZA
 PERIZZZZZZ PERZGIZOSA PERZGIZZZA PERZZIZOSA PERZZIZZZA
 PERZZZZZSA PERZZZZZZA PERZZZZZZA PESIGILOSA PESIGIZOSA
 PESIGIZZSA PESIGIZZZA PESIGZZOSA PESIGZZZZA PESIZZZZSA
 PESIZZZZSA PESIZZZZZA PEVIGIZOSA PEVIGIZZZA PEVIZZZOSA
 PEVIZZZZSA PEZBGIZOSA PEZBGIZZZA PEZBZZZOSA PEZBZZZZZA

PEZGGIZOSA PEZGGIZZSA PEZGGIZZZA PEZGGZZOSA PEZGGZZZZA
 PEZIBIZOSA PEZIBIZZZA PEZIBZZOSA PEZIBZZZZA PEZIGGZOSA
 PEZIGGZZZA PEZIGILOSA PEZIGIMOSA PEZIGINOSA PEZIGIROSA
 PEZIGIZOSA PEZIGIZOZA PEZIGIZZFA PEZIGIZZSA PEZIGIZZZA
 PEZIGIZZZZ PEZIGZZOSA PEZIGZZZFA PEZIGZZZSA PEZIGZZZZA
 PEZIGZZZZZ PEZIMIZOSA PEZIMIZZZA PEZIOZZOSA PEZIOZZZZA
 PEZISIZOSA PEZISIZZZA PEZIZILOSA PEZIZIZOSA PEZIZIZZSA
 PEZIZIZZZA PEZIZZIOSA PEZIZZZOSA PEZIZZZZFA PEZIZZZZSA
 PEZIZZZZZA PEZIZZZZZZ PEZQGIZOSA PEZQGIZZZA PEZQZZZOSA
 PEZQZZZZZA PEZZBIZOSA PEZZBIZZZA PEZZBZZOSA PEZZBZZZZA
 PEZZGGZOSA PEZZGGZZZA PEZZGICOSA PEZZGILOSA PEZZGIMOSA
 PEZZGINOSA PEZZGIROSA PEZZGIUOSA PEZZGIZOSA PEZZGIZOZA
 PEZZGIZZFA PEZZGIZZSA PEZZGIZZZA PEZZGIZZZZ PEZZGIZIOSA
 PEZZGZZOSA PEZZGZZZFA PEZZGZZZSA PEZZGZZZZA PEZZGZZZZZ
 PEZZQZZOSA PEZZQZZZSA PEZZQZZZZA PEZZZBZOSA PEZZZBZZA
 PEZZZILOSA PEZZZIMOSA PEZZZINOSA PEZZZIZOSA PEZZZIZZFA
 PEZZZIZZSA PEZZZIZZZA PEZZZIZZZZ PEZZZQZOSA PEZZZQZZA
 PEZZZZIOSA PEZZZZLOSA PEZZZZZBSA PEZZZZZBZA PEZZZZZGSA
 PEZZZZZGZA PEZZZZZOLA PEZZZZZOMA PEZZZZZORA PEZZZZZOSA
 PEZZZZZOSZ PEZZZZZOVA PEZZZZZQZA PEZZZZZQSA PEZZZZZQZA
 PEZZZZZZBA PEZZZZZZFA PEZZZZZZMA PEZZZZZZPA PEZZZZZZQA
 PEZZZZZZSA PEZZZZZZVA PEZZZZZZZA PEZZZZZZZB PEZZZZZZZH
 PEZZZZZZZM PEZZZZZZZQ PEZZZZZZZZ PFBIGIZOSA PFBIGIZZZA
 PZBIGILOSA PZBIGIZOSA PZBIGIZZSA PZBIGIZZZA PZBIGZZOSA
 PZBIGZZZSA PZBIGZZZA PZBIZIZOSA PZBIZIZZZA PZBIZZZOSA
 PZBIZZZZSA PZBIZZZZZA PZBZGIZOSA PZBZGIZZZA PZBZGZZOSA
 PZBZGZZZSA PZBZZIZOSA PZBZZIZZZA PZBZZZZOSA PZBZZZZZSA
 PZBZZZZZZA PZZIGILOSA PZZIGIZOSA PZZIGIZZSA PZZIGIZZZA
 PZZIGZZOSA PZZIGZZZSA PZZIGZZZZA PZZIZZZOSA PZZIZZZZSA
 PZZIZZZZZA PZZZGILOSA PZZZGIZOSA PZZZGIZZSA PZZZGIZZZA
 PZZZGZZOSA PZZZGZZZSA PZZZGZZZZA PZZZZILOSA PZZZZIZOSA
 PZZZZIZZSA PZZZZIZZZA PZZZZZZOSA PZZZZZZZSA PZZZZZZZZA
 PZZZZZZZZZ VEZZZZZOSA

E	E	E
GUATA	GUATA	GUATA QUATA
COSI	COST	<u>COST</u>
L	L	L
ANIMO	ANIMO	ANIMO
MIO	MIO	MIO
CHE	CHE	CHE
ANCOR	ANCQR	ANCOR
FUGGIVA	FUGSIVA	FUGGIVA FUGSIVA FUGSIZA FUGZIVA FUGZZVA FUGZZZA <u>FUOSIVA</u> FUSSIVA FUZSIVA FUZZIVA FUZZIZA FUZZZVA FUZZZZA FZGSIVA FZZSIVA FZZZIVA FZZZZVA FZZZZZA QUGSIVA QUGZIVA QUGZZVA QUOSIVA QUZSIVA QUZZIVA QUZZZVA QUZZZZA
SI	SV	SI <u>SV</u>
VOLSE	VOLSB	VOLBB VOLSB VOLSE VOLZB VOLZZ VOZZB VZZZB
A	A	A
RETRO	BEIRO	BBIRO <u>BEIRO</u> BZIRO
A	A	A
RIMIRAR	RIMIRAR	RIMIRAR
LO	LO	LO
PASSO	PASSO	PASSO
CHE	CHE	CHE
NON	NON	NON
LASCIO	LASCIO	LASCIO
GIA	GIR	<u>GIR</u>
MAI	MHI	CHI MAI MBI MHI MZI
PERSONA	PERSOMA	<u>PERSOMA</u>
VIVA	VIVR	VIBR VIOR <u>VIUR</u> VIVA VIVR VIZR VZZR

Bijlage C

Python-code

“The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.”

– E.W. Dijkstra

Deze bijlage bevat een implementatie van het EstiHMM-algoritme en het Viterbi-algoritme. De code voor het Viterbi-algoritme implementeert de algemene versie van paragraaf 3.4.3 en bevat een optie om de eenvoudiger versie van paragraaf 3.4.4 te gebruiken. Een deel van de code is gemeenschappelijk voor het Viterbi-algoritme en het EstiHMM-algoritme. Dat gedeelte van de code is ondergebracht in paragraaf C.3.

De code voor beide implementaties is geschreven in de programmeertaal Python. Ze is slechts summier gedocumenteerd. Ik geef ze hier enkel ter illustratie. Ze zal later in een volledig gedocumenteerde vorm online ter beschikking worden gesteld onder een Creative Commons License.

C.1 Viterbi-algoritme

```
class StationaryHMM(HMMalgoritme):  
  
    def __init__(self, marginal, transition, emission):  
  
        # this class is based on the class HMMalgorithm (used both  
        # by the Viterbi and EstiHMM algorithm)  
  
        HMMalgoritme.__init__(self)  
  
        self.nstates=len(marginal)  
        self.noutputs=len(emission[0])
```

```

self.marginallog=takelog(marginal)
self.transitionlog=takelog(transition)
self.emissionlog=takelog(emission)

self.maxproblog=[]

self.allPossible=[]

self.ties=False

# build a tree structure that contains all possible solutions
def BuildTreeWithAllPossibleSequences(self):

    for state in range(self.nstates):
        self.allPossible[self.nsteps-1].append(OptimalState(
            state))

    for i in [self.nsteps-1-j for j in range(1,self.nsteps)]:
        for state in range(self.nstates):
            node=OptimalState(state)
            for nextstate in range(self.nstates):
                node.addChild(self.allPossible[i+1][nextstate])
            self.allPossible[i].append(node)

def Viterbi(self, outputs, *args):

    """Determines the maximal sequence(s) for each subHMM and
    writes them to self.maximals.
    If *args == True, the algorithm will give all maximal
    sequences, if there are multiple.
    If *args == False, or if you do not specify a second
    argument, the algorithm will only give one arbitrary
    maximal sequence
    This method returns the execution time."""

    self.outputs=outputs
    self.nsteps=len(outputs)

    self.ties=False
    if len(args) is not 0:
        self.ties=args[0]

    self.maxproblog=[[0 for state in range(self.nstates)] for
        step in range(self.nsteps)]
    self.maxproblog[0]=[0]
    self.maximal=[[[] for state in range(self.nstates)] for step
        in range(self.nsteps)]
    self.maximal[0]=[[]]
    self.nmaximal=[[0 for state in range(self.nstates)] for step
        in range(self.nsteps)]
    self.nmaximal[0]=[0]

```

```

self.solutions=[[None for state in range(self.nstates)] for
                step in range(self.nsteps)]
self.solutions[0]=None
self.analysedSolutions=[[0 for state in range(self.nstates)]
                        for step in range(self.nsteps)]

self.allPossible=[[ ] for step in range(self.nsteps)]

self.BuildTreeWithAllPossibleSequences()

# Initialize timer
t0=time()

# If the length of the sequence is 1, we must consider this
  as a special case:
if self.nsteps == 1:
    max = self.marginallog[0]+self.emissionlog[0][outputs
        [0]]
    indexmax = [0]
    for state in range(1,self.nstates):
        prob = self.marginallog[state]+self.emissionlog[
            state][outputs[0]]
        if prob > max+TINY:
            max = prob
            indexmax =[state]
        elif self.ties == True:
            if prob >= max-TINY:
                indexmax.append(state)
    self.maxproblog[0]=max
    self.nmaximal[0]=len(indexmax)
    for i in range(len(indexmax)):
        self.maximal[0].append(OptimalState(indexmax[i]))
# If the length is 1, we can stop here, since we have
  found the maximal state(s).
self.TheAlgorithmHasRun()
return time()-t0

# We know now that the length of the sequence is greater
  than 1, so we must consider the following three steps:

# Find the optimal states for the final time
for previousstate in range(self.nstates):
    max = self.transitionlog[previousstate][0]+self.
        emissionlog[0][outputs[self.nsteps-1]]
    indexmax = [0]
    for state in range(1,self.nstates):
        prob = self.transitionlog[previousstate][state]+self
            .emissionlog[state][outputs[self.nsteps-1]]
        if prob > max+TINY:
            max = prob
            indexmax =[state]

```

```

        elif self.ties == True:
            if prob >= max-TINY:
                indexmax.append(state)
self.maxproblog[self.nsteps-1][previousstate]=max
self.nmaximal[self.nsteps-1][previousstate]=len(indexmax
)
for i in range(len(indexmax)):
    self.maximal[self.nsteps-1][previousstate].append(
        OptimalState(indexmax[i]))

# Find the optimal sequences by backwards recursion for all
# but the first step
# (will not be executed if the length of the sequence is 2)
for step in [self.nsteps-1-m for m in range(1, self.nsteps-1)
]:
    for previousstate in range(self.nstates):
        max = self.maxproblog[step+1][0]+self.transitionlog[
            previousstate][0]+self.emissionlog[0][outputs[
                step]]
        indexmax = [0]
        for state in range(1, self.nstates):
            prob = self.maxproblog[step+1][state]+self.
                transitionlog[previousstate][state]+self.
                emissionlog[state][outputs[step]]
            if prob > max+TINY:
                max = prob
                indexmax =[state]
            elif self.ties == True:
                if prob >= max-TINY:
                    indexmax.append(state)

self.maxproblog[step][previousstate]=max

nmax=0
for i in range(len(indexmax)):
    if (self.transitionlog[previousstate][indexmax[i]
        ]+self.emissionlog[indexmax[i]][outputs[step
        ]]) == float('-infinity'):
        self.maximal[step][previousstate].append(
            self.allPossible[step][indexmax[i]])
        nmax+=self.nstates**((self.nsteps-1-step))
    else:
        nmax+=self.nmaximal[step+1][indexmax[i]]
        node=OptimalState(indexmax[i])
        for nextnode in self.maximal[step+1][
            indexmax[i]]:
            node.addChild(nextnode)
        self.maximal[step][previousstate].append(
            node)
self.nmaximal[step][previousstate]=nmax

```

```

# Finish finding the optimal sequences by backwards
  recursion for the first step
max = self.maxproblog[1][0]+self.marginallog[0]+self.
  emissionlog[0][outputs[0]]
indexmax = [0]
for state in range(1,self.nstates):
  prob = self.maxproblog[1][state]+self.marginallog[state
    ]+self.emissionlog[state][outputs[0]]
  if prob > max+TINY:
    max = prob
    indexmax =[state]
  elif self.ties == True:
    if prob >= max-TINY:
      indexmax.append(state)

self.maxproblog[0]=max

nmax=0
for i in range(len(indexmax)):
  if (self.marginallog[indexmax[i]]+self.emissionlog[
    indexmax[i]][outputs[0]]) == float('-infinity'):
    self.maximal[0].append(self.allPossible[0][indexmax[
      i]])
    nmax+=self.nstates**((self.nsteps-1)
  else:
    nmax+=self.nmaximal[1][indexmax[i]]
    node=OptimalState(indexmax[i])
    for nextnode in self.maximal[1][indexmax[i]]:
      node.addChild(nextnode)
    self.maximal[0].append(node)
self.nmaximal[0]=nmax

self.TheAlgorithmHasRun()
# Return execution time
return time()-t0

def TheAlgorithmHasRun(self):
  self.algorithmhasrun = True
  print "The Viterbi algorithm has run succesfully"
  if self.ties:
    print "Ties were NOT resolved arbitrarily , unlike in the
      classical Viterbi algorithm"
  else:
    print "Ties were resolved arbitrarily , as in the
      classical Viterbi algorithm"

```

C.2 EstiHMM-algoritme

```

class BeliefModelForEstiHMM:
    """Simplification of BeliefModel with special properties for use
       in Hidden Markov Model prediction."""

    def __init__(self):
        self.nstates=0
        self.lmasslog=[]
        self.umasslog=[]
        self.threshold=[]

    def getLowerMassLog(self, state):
        return self.lmasslog[state]

    def getUpperMassLog(self, state):
        return self.umasslog[state]

    def getThresholdLog(self, optstate, otherstate):
        return self.thresholdlog[optstate][otherstate]

class ProbabilityIntervalBeliefModelForEstiHMM(BeliefModelForEstiHMM):
    """A subclass of BeliefModelForEstiHMM
       This subclass builds a model using only local lower and upper
       probabilities.
       More complicated belief models can easily be implemented, as
       long as they are subclasses of BeliefModelForEstiHMM."""

    def __init__(self, probabilityIntervals):
        BeliefModelForEstiHMM.__init__(self)
        self.nstates = len(probabilityIntervals)
        lmass = [interval[0] for interval in probabilityIntervals]
        umass = [interval[1] for interval in probabilityIntervals]
        self.lmasslog = takelog(lmass)
        self.umasslog = takelog(umass)
        threshold = [[float(lmass[otherstate])/umass[optstate] for
                       otherstate in range(self.nstates)] for optstate in range(
                       self.nstates)]
        self.thresholdlog = takelog(threshold)

    def giveProbabilityIntervalBeliefModelForEstiHMM(
        probabilityIntervals):
        return ProbabilityIntervalBeliefModelForEstiHMM(
            probabilityIntervals)

    def giveConditionalProbabilityIntervalBeliefModelForEstiHMM(
        listOfProbabilityIntervals):
        return [ProbabilityIntervalBeliefModelForEstiHMM(
            ProbabilityIntervals) for ProbabilityIntervals in
            listOfProbabilityIntervals]

```

```

class StationaryHMM(HMMalgoritme):

    def __init__(self, marginalBM, transitionBM, emissionBM):

        # marginalBM: instance of BeliefModelForEstiHMM
        # transitionBM: list of instances of BeliefModelForEstiHMM
        # emissionBM: list of instances BeliefModelForEstiHMM

        HMMalgoritme.__init__(self)

        self.nstates=marginalBM.nstates
        self.noutputs=emissionBM[0].nstates

        self.marginal=marginalBM
        self.transition=transitionBM
        self.emission=emissionBM

        self.alphamaxlog=[]
        self.alphaminlog=[]
        self.betamaxlog=[]
        self.alphaoptlog=[]

        self.allPossible=[]

        self.stepcount=0
        self.previousstatecount=0

    def DetermineBetaMaxLog(self, outputs):
        """Does a Bellman-like recursion for the logarithms of the
        BetaMax, and returns a nested list with their values."""

        # Initialise the BetaMaxLog function on states for the final
        time
        for state in range(self.nstates):
            self.betamaxlog[self.nsteps-1].append(self.emission[
                state].getLowerMassLog(outputs[self.nsteps-1]))

        # For each previous time, do a maximisation
        for step in [self.nsteps-1-m for m in range(1, self.nsteps)]:
            for state in range(self.nstates):
                tomaximise=[]
                for nextstate in range(self.nstates):
                    tomaximise.append(self.transition[state].
                        getLowerMassLog(nextstate)+self.betamaxlog[
                            step+1][nextstate])
                self.betamaxlog[step].append(max(tomaximise)+self.
                    emission[state].getLowerMassLog(outputs[step]))

        # Return the sequence of BetaMaxLog functions
        return self.betamaxlog

```

```

def DetermineAlphaMaxLog(self, outputs):
    """Does a Bellman-like recursion for the logarithms of the
        AlphaMax, and returns a nested list with their values."""

    # Initialise the AlphaMaxLog function on states for the
    # final time
    for state in range(self.nstates):
        self.alphamaxlog[self.nsteps-1].append(self.emission[
            state].getUpperMassLog(outputs[self.nsteps-1]))

    # For each previous time, do a maximisation
    for step in [self.nsteps-1-m for m in range(1, self.nsteps)]:
        for state in range(self.nstates):
            tomaximise=[]
            for nextstate in range(self.nstates):
                tomaximise.append(self.transition[state].
                    getUpperMassLog(nextstate)+self.alphamaxlog[
                        step+1][nextstate])
            self.alphamaxlog[step].append(max(tomaximise)+self.
                emission[state].getUpperMassLog(outputs[step]))

    # Return the sequence of AlphaMaxLog functions
    return self.alphamaxlog

def DetermineAlphaMinLog(self, outputs):
    """Does a Bellman-like recursion for the logarithms of the
        AlphaMin, and returns a nested list with their values."""

    # Initialise the AlphaMinLog function on states for the
    # final time
    for state in range(self.nstates):
        self.alphaminlog[self.nsteps-1].append(self.emission[
            state].getUpperMassLog(outputs[self.nsteps-1]))

    # For each previous time, do a minimisation
    for step in [self.nsteps-1-m for m in range(1, self.nsteps)]:
        for state in range(self.nstates):
            tominimise=[]
            for nextstate in range(self.nstates):
                tominimise.append(self.transition[state].
                    getUpperMassLog(nextstate)+self.alphaminlog[
                        step+1][nextstate])
            self.alphaminlog[step].append(min(tominimise)+self.
                emission[state].getUpperMassLog(outputs[step]))

    # Return the sequence of AlphaMaxLog functions
    return self.alphaminlog

def DetermineAlphaOptLog(self):
    """Calculates the logarithms of the AlphaOpt.

```



```

Assigns a list , for each time step , of the successive values
    (conditional on previous state) to self.alphaoptlog.
Returns self.alphaoptlog. """

# For all times but the initial time
for step in range(1,self.nsteps):
    for previousstate in range(self.nstates):
        for optstate in range(self.nstates):
            tomaximise=[]
            for otherstate in range(self.nstates):
                if otherstate!=optstate:
                    tomaximise.append(self.transition[
                        previousstate].getThresholdLog(
                            optstate , otherstate)+self.betamaxlog[
                                step][ otherstate ])
            self.alphaoptlog[step][previousstate].append(max
                (tomaximise))

# For the initial time
self.alphaoptlog[0]=[]
for optstate in range(self.nstates):
    tomaximise=[]
    for otherstate in range(self.nstates):
        if otherstate!=optstate:
            tomaximise.append(self.marginal.getThresholdLog(
                optstate , otherstate)+self.betamaxlog[0][
                    otherstate ])
    self.alphaoptlog[0].append(max(tomaximise))

# Return the sequence of AlphaOptLog functions
return self.alphaoptlog

def BuildTreeWithAllPossibleSequences(self):

    for state in range(self.nstates):
        self.allPossible[self.nsteps-1].append(OptimalState(
            state))

    for i in [self.nsteps-1-j for j in range(1,self.nsteps)]:
        for state in range(self.nstates):
            node=OptimalState(state)
            for nextstate in range(self.nstates):
                node.addChild(self.allPossible[i+1][nextstate])
            self.allPossible[i].append(node)

def EstiHMM(self , outputs):
    """Determines the maximal sequences and writes them to self.
    maximals.
    Returns the execution time."""

```

```

self.outputs=outputs
self.nsteps=len(outputs)

self.maximal=[[[] for state in range(self.nstates)] for step
              in range(self.nsteps)]
self.maximal[0]=[]
self.nmaximal=[[0 for state in range(self.nstates)] for step
              in range(self.nsteps)]
self.solutions=[[None for state in range(self.nstates)] for
               step in range(self.nsteps)]
self.solutions[0]=None
self.analysedSolutions=[[0 for state in range(self.nstates)]
                       for step in range(self.nsteps)]

self.alphamaxlog=[[[] for step in range(self.nsteps)]
                 for state in range(self.nstates)]
self.alphaminlog=[[[] for step in range(self.nsteps)]
                 for state in range(self.nstates)]
self.betamaxlog=[[[] for step in range(self.nsteps)]
                for state in range(self.nstates)]

self.alphaoptlog=[[[] for state in range(self.nstates)] for
                 step in range(self.nsteps)]

self.allPossible=[[[] for step in range(self.nsteps)]
                 for state in range(self.nstates)]
self.BuildTreeWithAllPossibleSequences()
self.DetermineAlphaMaxLog(outputs)
self.DetermineAlphaMinLog(outputs)
self.DetermineBetaMaxLog(outputs)
self.DetermineAlphaOptLog()

# Initialize timer
t0=time()

# If the length of the sequence is 1, we must consider this
# as a special case:
if self.nsteps == 1:
    for state in range(self.nstates):
        if self.alphamaxlog[0][state]>=self.alphaoptlog[0][
            state]-TINY:
            self.maximal[0].append(OptimalState(state))
            self.nmaximal[0][0]+=1
    # If the length is 1, we can stop here, since we have
    # found the maximal state(s).
    self.TheAlgorithmHasRun()
    return time()-t0

# We know now that the length of the sequence is greater
# than 1, and so we must consider the following three steps

# Find the optimal states for the final time
for previousstate in range(self.nstates):
    for state in range(self.nstates):
        if self.alphamaxlog[self.nsteps-1][state]>=self.

```

```

alphaoptlog [ self . nsteps - 1 ][ previousstate ][ state ] -
TINY:
    self . maximal [ self . nsteps - 1 ][ previousstate ].
        append ( OptimalState ( state ))
    self . nmaximal [ self . nsteps - 1 ][ previousstate ] += 1

# Find the optimal sequences by backwards recursion for all
# but the first step
for step in [ self . nsteps - 1 - m for m in range ( 1 , self . nsteps - 1 )
]:
# ( will not be executed if the length of the sequence is 2 )
    self . stepcount = step
    for previousstate in range ( self . nstates ):
        self . previousstatecount = previousstate
        for state in range ( self . nstates ):
            if self . alphaminlog [ step ][ state ] >= self .
alphaoptlog [ step ][ previousstate ][ state ] - TINY:
                if ( self . transition [ previousstate ].
                    getLowerMassLog ( state ) + self . emission [
                    state ]. getLowerMassLog ( outputs [ step ] ) ) ==
                    float ( '-infinity ' ):
                    self . maximal [ step ][ previousstate ]. append
                        ( self . allPossible [ step ][ state ] )
                    self . nmaximal [ step ][ previousstate ] += self
                        . nstates ** ( self . nsteps - 1 - step )
            else :
                node = OptimalState ( state )
                for nextstate in self . maximal [ step + 1 ][
                    state ]:
                    node . addChild ( nextstate )
                self . maximal [ step ][ previousstate ]. append
                    ( node )
                self . nmaximal [ step ][ previousstate ] += self
                    . nmaximal [ step + 1 ][ state ]
            elif self . alphamaxlog [ step ][ state ] >= self .
alphaoptlog [ step ][ previousstate ][ state ] - TINY:
                self . maximal [ step ][ previousstate ]. append (
                    OptimalState ( state ) )
                if ( self . transition [ previousstate ].
                    getLowerMassLog ( state ) + self . emission [
                    state ]. getLowerMassLog ( outputs [ step ] ) ) ==
                    float ( '-infinity ' ):
                    possibleSequences = self . allPossible [
                        step + 1 ]
            else :
                possibleSequences = self . maximal [ step
                    + 1 ][ state ]
    self . PruneNonOptimalBranches ( self . maximal [
        step ][ previousstate ][ - 1 ],
        possibleSequences , step , self . alphaoptlog [
        step ][ previousstate ][ state ] )

```

```

self.stepcount=0
self.previousstatecount=0
for state in range(self.nstates):
    if self.alphaminlog[0][state]>=self.alphaoptlog[0][state]-TINY:
        if (self.marginal.getLowerMassLog(state)+self.emission[state].getLowerMassLog(outputs[0])) == float('-infinity'):
            self.maximal[0].append(self.allPossible[0][state])
            self.nmaximal[0][0]+=self.nstates**(self.nsteps-1)
        else:
            node = OptimalState(state)
            for nextstate in self.maximal[1][state]:
                node.addChild(nextstate)
            self.maximal[0].append(node)
            self.nmaximal[0][0]+=self.nmaximal[1][state]
    elif self.alphamaxlog[0][state]>=self.alphaoptlog[0][state]-TINY:
        self.maximal[0].append(OptimalState(state))
        if (self.marginal.getLowerMassLog(state)+self.emission[state].getLowerMassLog(outputs[0])) == float('-infinity'):
            possibleSequences = self.allPossible[1]
        else:
            possibleSequences = self.maximal[1][state]
        self.PruneNonOptimalBranches(self.maximal[0][-1], possibleSequences,0,self.alphaoptlog[0][state])

self.nmaximal[0]=self.nmaximal[0][0]

self.TheAlgorithmHasRun()
# Return execution time
return time()-t0

def PruneNonOptimalBranches(self,node,nextnodes,step,threshold):
    """Performs the forward part of the backward-forward recursion"""
    state = node.index
    if step==self.nsteps-2:
        for nextnode in nextnodes:
            nextstate = nextnode.index
            factor=self.transition[state].getUpperMassLog(nextstate)+self.emission[state].getUpperMassLog(self.outputs[step])
            if self.alphamaxlog[step+1][nextstate]+factor >= threshold-TINY:
                nextoptimalstate=OptimalState(nextstate)
                node.addChild(nextoptimalstate)

```

```

        self.nmaximal[ self.stepcount ][ self.
            previousstatecount ] += 1
    else :
        for nextnode in nextnodes :
            nextstate = nextnode.index
            factor = self.transition[ state ].getUpperMassLog(
                nextstate) + self.emission[ state ].getUpperMassLog(
                self.outputs[ step ])
            if self.alphamaxlog[ step + 1 ][ nextstate ] + factor >=
                threshold - TINY :
                nextoptimalstate = OptimalState( nextstate )
                self.PruneNonOptimalBranches( nextoptimalstate ,
                    nextnode.getChildren() , step + 1 , threshold -
                    factor )
                node.addChild( nextoptimalstate )

def TheAlgorithmHasRun( self ) :
    self.algorithmhasrun = True
    print "The EstiHMM algorithm has run succesfully"

```

C.3 Gemeenschappelijke code

```

import math
from time import time
from types import ListType

TINY = 1e-40

def takelognumber(number):
    if number < -TINY:
        print "you cannot take the logarithm of a negative number"
    elif number < TINY:
        return float('-infinity')
    else:
        return math.log(float(number))

def takelog(notlog):
    if not isinstance(notlog, ListType):
        return takelognumber(notlog)
    else:
        log = []
        for i in range(len(notlog)):
            log.append(takelog(notlog[i]))
        return log

def makenumbers(listOfNumbersAsStrings):
    return [int(number) for number in listOfNumbersAsStrings]

```

```
class OptimalState:
    """Class for nodes used for recursive construction of a tree of
    optimal states."""

    def __init__(self, index):
        self.index=index
        self.children=[]

    def addChild(self, state):
        self.children.append(state)

    def getChildren(self):
        return self.children

class HMMalgoritme:

    def __init__(self):

        self.nsteps=0
        self.nstates=0
        self.noutputs=0

        self.maximal=[]
        self.nmaximal=[]

        self.solutions=[]
        self.algorithmhasrun = False

        self.outputs=[]
        self.ties=True

    def itHasNotRunYet(self):
        print "you need to run the algorithm first!"

    def getMaximalSequences(self):
        #returns the maximal sequences of all subHMMs in the form of
        #treestructures (which is how they come out of the
        #algorithm)
        if not self.algorithmhasrun:
            self.itHasNotRunYet()
        else:
            return self.maximal

    def getListOfSolutions(self, index, *args):
        # Turns the tree structure of maximal sequences for a given
        # subHMM into a list of solutions
        # *args can hold the previous state, if index is larger than
        # zero
        # if you do not specify a previous state, the
        # listOfSolutions for all previous states is determined
        TreeOfMaximals=[]
```

```

if not self.algorithmhasrun:
    self.itHasNotRunYet()
elif index == 0:
    if self.solutions[0]==None:
        TreesOfMaximals=self.maximal[0]
        self.solutions[0]=self.TurnTreesIntoList(
            TreesOfMaximals)
    return self.solutions[0]
else:
    previousstates=range(self.nstates)
    if len(args)is not 0:
        previousstates=[args[0]]

    for previousstate in previousstates:
        if self.solutions[index][previousstate]==None:
            TreesOfMaximals=self.maximal[index][
                previousstate]
            self.solutions[index][previousstate]=self.
                TurnTreesIntoList(TreesOfMaximals)

    if len(args)==0:
        return self.solutions[index]
    else:
        return self.solutions[index][args[0]]

def TurnTreesIntoList(self ,TreesOfMaximals):
    solutions=[]
    solutionUpTillNow=""
    self.TurnTreesIntoListRecursive(TreesOfMaximals ,
        solutionUpTillNow , solutions)
    solutionsAsLists = [makenumbers(solution.split("|")) for
        solution in solutions]
    return solutionsAsLists

def TurnTreesIntoListRecursive(self ,TreesOfMaximals ,
    solutionUpTillNow , solutions):
    for node in TreesOfMaximals:
        newSolutionUpTillNow=solutionUpTillNow+str(node.index)+"
            |"
        if len(node.getChildren())==0:
            solutions.append(newSolutionUpTillNow[:-1])
        else:
            self.TurnTreesIntoListRecursive(node.getChildren() ,
                newSolutionUpTillNow , solutions)

def getAllListsOfSolutions(self):
    for i in range(self.nsteps):
        self.getListOfSolutions(i)
    return self.solutions

def printListOfSolutions(self , list):

```

```

for solution in list:
    string=""
    for state in solution:
        string+=str(state)+"|"
    string=string[:-1]
    print string

def printMaximalSequences(self ,index ,* args):
    # *args can hold the previous state , if index is larger than
    zero
    # if you do not specify a previous state , the maximal
    sequences for all previous states are printed

    if (index == 0) and (self.nsteps is not 1) :
        self.getListOfSolutions(0)
        print "List of maximal sequences x(1:"+str(self.nsteps)+
        ")"
        print "Total number = "+str(self.nmaximal[0])
        print "("+str(100*float(self.nmaximal[0])/float(self.
        nstates**self.nsteps))+"% of possible number)"
        self.printListOfSolutions(self.solutions[0])

    else :
        if index == self.nsteps-1:
            print "List of maximal sequences x("+str(self.nsteps
            )+")"
        else :
            print "List of maximal sequences x("+str(index+1)+":
            "+str(self.nsteps)+")"
        previousstates=range(self.nstates)
        if len(args) is not 0:
            previousstates=[args[0]]
            self.getListOfSolutions(index , args[0])
        else :
            self.getListOfSolutions(index)

        for previousstate in previousstates:
            print "previous state = "+str(previousstate)
            print "Total number = "+str(self.nmaximal[index][
            previousstate])
            self.printListOfSolutions(self.solutions[index][
            previousstate])

def printAllMaximalSequences(self):
    for i in range(self.nsteps):
        self.printMaximalSequences(i)

def printNumberOfMaximalSequences(self ,index ,* args):
    # *args can hold the previous state , if index is larger than
    zero
    # if you do not specify a previous state , the number of

```



```
        maximal states for all previous states are printed

if not self.algorithmhasrun:
    self.itHasNotRunYet()
    return None

if (index == 0) and (self.nsteps is not 1) :
    print "Number of maximal sequences x(1:"+str(self.nsteps
        )+"): "+str(self.nmaximal[0])
    print "("+str(100*float(self.nmaximal[0])/float(self.
        nstates**self.nsteps))+ "% of possible number)"

else :
    if index == self.nsteps-1:
        print "Number of maximal sequences x("+str(self.
            nsteps)+")"
    else :
        print "Number of maximal sequences x("+str(index+1)+
            ":"+str(self.nsteps)+")"
    previousstates=range(self.nstates)
    if len(args)is not 0:
        previousstates=[args[0]]

    for previousstate in previousstates :
        print "previous state = "+str(previousstate)
        print "Total number = "+str(self.nmaximal[index][
            previousstate])

def printAllNumbersOfMaximalSequences(self):
    for i in range(self.nsteps):
        self.printNumberOfMaximalSequences(i)

# That's all folks
```

Referentielijst

- [1] Jasper De Bock and Gert de Cooman. State sequence prediction in imprecise hidden markov models. In *ISIPTA '11 – Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications*, 2011. To appear.
- [2] Peter Walley. *Statistical reasoning with imprecise probabilities*. Chapman and Hall, London, first edition, 1991.
- [3] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains*. Undergraduate Text in Mathematics. Springer-Verlag, New York, 1976. Reprint of the 1960 Edition.
- [4] Lawrence R. Rabiner. A tutorial on HMM and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- [5] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [6] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2): 260–269, 1967. doi: 10.1109/TIT.1967.1054010.
- [7] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.
- [8] Gert de Cooman, Filip Hermans, Alessandro Antonucci, and Marco Zaffalon. Epistemic irrelevance in credal networks: the case of imprecise Markov trees. In *ISIPTA '09 – Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications*, pages 149–158, 2009.
- [9] Fabio G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.
- [10] Fabio G. Cozman. Graphical models for imprecise probabilities. *International Journal of Approximate Reasoning*, 39(2-3):167–184, June 2005. doi: doi:10.1016/j.ijar.2004.10.003.
- [11] Gert de Cooman, Enrique Miranda, and Marco Zaffalon. Independent natural extension. *Artificial Intelligence*, 2010. Accepted for publication.

-
- [12] Inés Couso, Serafín Moral, and Peter Walley. A survey of concepts of independence for imprecise probabilities. *Risk Decision and Policy*, 5:165–181, 2000.
- [13] Gert de Cooman, Enrique Miranda, and Marco Zaffalon. Independent natural extension. In Eyke Hüllermeier, Rudolf Kruse, and Frank Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design (Proceedings of IPMU 2010, 28 June – 2 July 2010, Dortmund, Germany)*, volume 6178 of *Lecture Notes in Computer Science*, pages 737–746. Springer, Heidelberg, 2010.
- [14] Enrique Miranda and Gert de Cooman. Marginal extension in the theory of coherent lower previsions. *International Journal of Approximate Reasoning*, 46(1):188–225, September 2007. doi: 10.1016/j.ijar.2006.12.009.
- [15] Matthias C. M. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45(1):17–29, January 2007. doi: 10.1016/j.ijar.2006.06.001.
- [16] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957.
- [17] Luis M. de Campos, Juan F. Huete, and Serafín Moral. Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2:167–196, 1994.
- [18] Gert de Cooman, Matthias C. M. Troffaes, and Enrique Miranda. n -Monotone exact functionals. *Journal of Mathematical Analysis and Applications*, 347:143–156, 2008.
- [19] Peter Walley. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 58:3–57, 1996. With discussion.