Visualisierung kontinuierlicher, multimodaler Schmerz Scores am Beispiel akustischer Signale

Masterarbeit

Franz Anders
HTWK Leipzig

Januar 2017

Abstract

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

2 Grundlagen der Schmerzbewertung mit Hilfe akustischer Signale

Das Ziel dieses Kapitels ist es, wichtig Grundlagen zum Verständnis der Schmerzbewertung bei Neugeborenen auf Basis akustischer Signale zu legen. Dazu wird in Kapitel 2.1 zunächst erläutert, wie die Schmerzbewertung aus Sicht medizinischer Fachkräfte im klinischen Alltag durchgeführt wird. Der Fokus liegt dabei insbesondere auf der Ableitung des Schmerzgrades aus dem Weinen des Babys. Um die Stimme des Babys automatisiert analysieren zu können, werden Methoden der Signalverarbeitung verwendet. Daher werden in Kapitel 2.2 Grundlagen zur akustischen Modellierung der menschlichen Stimme erläutert. In Kapitel 2.3 wird eine Einführung in die „klassische Schreiforschung“ gegeben. Dabei handelt es sich um Wissenschaftsgebiet, bei dem Methoden der Signalverarbeitung verwendet werden, um tiefergehende Analysen des Weinens Neugeborener durchzuführen. Da sich das in dieser Arbeit vorgestellte Konzept als Erweiterung der klassischen Methoden versteht, ist ein Verständnis dieses Wissenschaftsgebietes unerlässlich. In Kapitel 2.4 werden Grundlagen des überwachten maschinellen Lernens erläutert, da diese bei der Spracherkennung ausgiebigen Einsatz erfahren.

2.1 Schmerz und Weinen bei Neugeborenen aus medizinischer Sicht

Schmerz wird definiert als eine „ein unangenehmes Sinnes- oder Gefühlserlebnis, das mit tatsächlicher oder potenzieller Gewebeschädigung einhergeht“.[38, S. 438] Abseits von dieser theoretischen Definition hat der Mensch ein intuitives Verständnis für Schmerz, da jeder ihn in seine Leben mindestens Einmal erfahren musste. In der ersten Hälfte des 20ten Jahrhunderts war die vorherrschende Meinung, dass Neugeborene keinen Schmerz empfinden können. Beispielsweise bekamen sie nach Operationen keine Schmerzmittel verabreicht. Der aktuelle Stand der Forschung besagt, dass Neugeborene im selben Maße wie Erwachsene Schmerz empfinden können. Die freien Nervenenden, die in der Lage sind, physische Schäden am Körper festzustellen, sind bei Neugeborenen ebenso wie bei Erwachsenen über den Körper verteilt. Die hormonelle Reaktion ist ebenfalls vergleichbar. [18, S. 402] [38, S. 438]

2.1.1 Pain Scales

Es gibt diverse Gründe, die bei Neugeborenen Schmerz verursachen können. Sie reichen über physische Schäden, aufgrund von Komplikationen bei der Geburt oder Gewalteinwirkungen, über Erkrankungen, wie Kopfschmerzen oder Infektionen, bis hin zu therapeutischen Prozeduren, wie Injektionen oder Desinfektionen von Wunden. Das Vorhandensein von Schmerz ist anhand diverser physiologischer, biochemischer, verhaltensbezogener und psychologischer

Veränderungen messbar. Die für diese Arbeit wichtigste Verhaltensänderung ist das Weinen, welches zu den verhaltensbezogenen Veränderungen gezählt wird.[38, S. 441]

Schlussendlich ist Schmerz ein subjektives Empfinden. Das heißt, dass ein und der selbe Stimulus bei zwei verschiedenen Personen zu einem unterschiedlichem Schmerzempfinden führen kann. Daher wird der Schmerzgrad bei Erwachsenen typischerweise durch eine Selbsteinschätzung des Patienten unter der Leitung gezielter Fragen des Arztes festgestellt. Bei Kindern unter 3 Jahren ist diese Selbsteinschätzung nicht möglich. Diese Einschätzung muss daher von anderen Personen vorgenommen werden. Im klinischen Kontext sind dies medizinische Fachkräfte, wie beispielsweise Ärzte, Krankenpfleger oder Geburtshelfer. Die von außen am leichtesten feststellbaren Indikatoren von Schmerz sind die verhaltensbasierten Merkmale, wie zum Beispiel ein Verkrampfen des Gesichtsausdrucks, erhöhte Körperbewegungen oder lang anhaltendes Weinen.[38, S. 438] Die Schmerzdiagnostik durch eine andere Person ist, genau wie das Schmerzempfinden, etwas inherent subjektives und abhängig von Faktoren wie dem Alter, Geschlecht, kulturellen Hintergrund, persönlichen Erfahrungen mit Schmerz usw.[13, S. 3] Um die Schmerzdiagnostik objektiver zu gestalten, wurden daher sogenannte *Pain Scales* entwickelt, welche mit Hilfe eines Punktesystems den Schmerzgrad des Babys quantifizieren.[38, S. 438 - 439] Es existieren *monomodale / unidimensionale* Pain Scales, bei denen der Schmerzgrad aus der Beobachtung *eines* Merkmals geschlossen wird, wie beispielsweise des Gesichtsausdrucks. Ein Merkmal wird in diesem Zusammenhang als *Schmerzindikator* bezeichnet. *Multimodale / Multidimensionale* Pain Scales beziehen mehrere Schmerzindikatoren in das Scoring mit ein.[23, S. 69 - 71].

Tabelle 2.1 zeigt das Scoring-System „Neonatal Infant Pain Scale“ (NIPS) als Beispiel für eine multimodale Pain Scale. Diese Pain Scale ist für Babys von 0 bis 1 Jahr geeignet. Sie ist vor allem für die Diagnose schon Schmerz geeignet, der während einer Prozedur entsteht. Sie wurde auf der Basis der Erfahrungen von Krankenschwestern erarbeitet. Das Baby soll bei der Anwendung dieser Pain Scale für ungefähr eine Minute beobachtet werden. Wird die Scale verwendet, um den Schmerzgrad nach der Prozedur festzustellen, wird Empfohlen, die Diagnose alle 30 Minuten durchzuführen. Für jede der aufgeführte Kategorie (Schmerzindikatoren) werden ein, zwei oder drei Punkte vergeben und anschließend aufsummiert. Ein insgesamter Wert von > 3 zeigt moderaten Schmerz an, ein Wert von > 4 großen Schmerz.[15] [35, S. 98]

Tabelle 2.1: Neonatal Infant Pain Scale (NIPS) [15]

NIPS	**0 points**	**1 point**	**2 points**
Facial Expr.	Relaxed	Contracted	-
Cry	Absent	Mumbling	Vigorous
Breathing	Relaxed	Different than basal	-
Arms	Relaxed	flexed/stretched	-
Legs	Relaxed	flexed/stretched	-
Alertness	Sleeping	uncomfortable	-

Nach dem Muster der NIPS existieren viele weitere Pain Scales. Sie unterscheiden sich hinsichtlich der Schmerzindikatoren, die betrachtet werden, dem Punktesystem, der Art des Schmerzes, die festzustellen ist, dem Beobachtungszeitraum usw. Einige Pain Scales sind beispielsweise auf die Schmerzdiagnostik während eines Eingriffes spezialisiert, andere auf den darauf folgenden Heilungsprozess. In den meisten multimodalen Pain Scales wird das Weinen oder Schreien der Babys als Schmerzindikator mit einbezogen. In der englischen Fachliteratur ist von „Cry“ die Rede.[35, S. 97 - 98] In dieser Arbeit wird „Cry“ mit „Weinen“

oder mit dem neutraleren Begriff „kindliche Lautäußerungen“ übersetzt. Tabelle 2.2 zeigt eine Übersicht über einige multimodale Pain Scales. Die Übersicht zeigt vor allem, nach welchen Kriterien das Weinen in den jeweiligen Scales bewertet wird. Außerdem wird für jede Pain Scale angegeben, für welches Alter sie geeignet ist, welcher Schmerz-Typ diagnostiziert wird, sowie der zur Diagnose vorgesehene Beobachtungszeitraum und -Intervall. Angaben, die mit einem *?* verzeichnet wurden, konnten nicht in Erfahrung gebracht werden. Es handelt sich hierbei nur eine Übersicht über die wichtigsten Fakten der Pain Scales. Die Anleitungen der jeweiligen Pain Scales geben weitere Anweisungen zur Benutzung.

System	**P.**	**Description**	**other Ind.**	**Comments**
FLACC	0	No cry (awake or asleep)	Face, Legs, Activity, Consolability	Age: 2 months - 7 years Observe for: 1 - 5 minutes Observe every: ? Pain-Type: Ongoing
	1	Moans or whimpers; occasional complaint		
	2	Crying steadily, screams or sobs, frequent complaints		
N-PASS	-2	No cry with painful stimul	Behaviour, Facial Expr., Extremities, Vital Signs	Age: 0 - 100 days Observe for: ? Observe every: 2 - 4 hours Pain-Type: Ongoing
	-1	Moans or cries minimally with painful stimuli		
	0	Appropiate Crying		
	1	Irritable or Crying at Intervals. Consolable		
	2	High-pitched or silent-continuous crying. Not consolable		
BPSN	0	No Crying	Alertness, Skin Color, Eyebrows, ...	Age: ? Observe for: ? Observe every: Pain Type: ?
	1	Crying less than 2 minutes		
	2	Crying more than 2 minutes		
	3	Shrill Crying more than 2 minutes		
CRIES	0	If no cry or cry which is not high pitched	O2, Vital Signs, Expression, Sleeplessness	Age: 0 - 6 Months Observe for: ? Observe every: 1 hour Pain-Type: Post Operative
	1	If cry high pitched but baby. is easily consoled		
	2	If cry is high pitched and baby is inconsolable		
COVERS	0	No Cry	O2, Vital Signs, Expression, ...	Age: ? Observe for: ? Observe every: ? Pain Type: Procedural
	1	High-Pitched or visibly crying		
	2	Inconsolable or difficult to soothe		
PAT	0	No	Posture, Sleep Pattern, Expression, ...	Age: 0 - 3 months Observe for: 15 - 30 sec Observe every: 30 min Pain Type: Post Operative
	1	When disturbed, doesn't settle after handling, loud, whimper, whining		
DAN	0	Moans Briefly	Facial Exp., Limb Mov.	Age: 0 - 2 years Observe for: ? Observe every: ? Pain Type: Procedural
	1	Intermittent Crying		
	2	Long-Lasting Crying, Continuous howl		
COMFORT	0	No crying	Alertness, Calmness, Respiration, ...	Age: 0 - 3 years Observe for: ? Observe every: ? Pain: Post Operative
	1	Sobbing or gasping		
	2	Moaning		
	3	Crying		
	4	Screaming		
MBPS	0	Laughing or giggling	Facial Exp., Movement	Age: ?

1	Not Crying	Observe for: ? Observe every: ?
2	Moaning quiet vocalizing gentle or whimpering cry	Pain Type: Procedural
3	Full lunged cry or sobbing	
4	Full lunged cry more than baseline cry	

Tabelle 2.2: Übersicht über Pain-Scales. [35, S. 98] [45] [39] [9] [3] [19] [16] [5] [34] [9]

Da die Begriffe *Pain Scale* und *Pain Score* in einigen Veröffentlichungen inkonsistent verwendet werden, wird in dieser Arbeit die Konvention getroffen, dass mit *Pain Scale* das System zur Schmerzdiangostik gemeint ist und mit *Pain Score* die auf Basis der Pain Scale vergebene Punktzahl. *NIPS* ist also beispielsweise eine Pain Scale, und 3 eine Pain Score.

Die folgende Schlussfolgerungen werden bezüglich der Pain Scales aus Tabelle 2.2 gezogen:

1. Die Kriterien zur Bewertung des Weinens werden zum größten Teil mit *subjektiv behafteten Begriffen* beschrieben. Beispielsweise wird bei dem *N-PASS*-System ein Score von drei für „High-pitched or silent-continuous crying“ vergeben. Die Begriffe „high-pitched“ und „silent-continuous“ werden nicht näher definiert. Auch die Anwendungsvorschriften der Pain Scales geben keine festen Definitionen. Dies erleichtert den praktischen Einsatz der Pain Scales, führt jedoch zu einem Interpretationsspielraum und somit zu einem von der diagnostizierenden Person abhängigen Scoring. Die *BPSN*-Scale nutzt als einzige der vorgestellten Scales objektiv messbare Eigenschaften.
2. Die Pain Scales fokussieren unterschiedliche Eigenschaften zur Bestimmung der Pain Score bezüglich des Weinen-Indikators. Bei *CRIES* ist die Tonhöhe, bei *BPSN* die Länge und bei *COMFORT* die Art des Weinens ausschlaggebend für ein höheres Scoring.
3. Die Beschreibungen sind kurz und prägnant gehalten, die diagnostizierende Person hat bei keiner Pain Scale auf mehr als drei Eigenschaften des Weinens zu achten.

2.1.2 Weinen bei Neugeborenen

An dieser Stelle stellt sich der Leser eventuell die Frage, woher die unterschiedlichen Bewertungskriterien für das Weinen in den Pain Scales stammen. Gibt es eine „beste“ Pain Scale? Dieser Frage unterliegen zwei grundlegendere Fragen:

1. Ist es möglich, aus den akustischen Eigenschaften den motivierenden Grund für die Lautäußerung abzuleiten? Klingt ein durch Hunger bedingtes Weinen anders als ein durch Schmerz bedingtes?
2. Ist es möglich, anhand der akustischen Eigenschaften den Schweregrad dieses motivierenden Grundes abzuleiten?

Die Annahme, dass es möglich sei, aus den Eigenschaften des Weinens den Grund ablesen zu können, wird als „Cry-Types Hypothesis“ bezeichnet. Die berühmtesten Befürworter dieser Hypothese ist eine skandinavische Forschungsgruppe, auch bezeichnet als „Scandinavian Cry-Group“, die die Idee in dem Buch „Infant Crying: Theoretical and Research Perspectives“[26] publik machte. Die Hypothese besagt, dass die Empfindungen *Hunger, Freude, Schmerz, Geburt* sowie Sonstiges klare Unterschiede hinsichtlich der akustischen Merkmale des

Weinens aufweisen würden. Diese Unterschiede seien im Spektogramm sichtbar (Siehe Kapitel 2.2.4). Wenige Jahre Später zeigten Müller et al. [8], dass bei leichter Veränderung der Experimentbedingungen die Unterscheidung nicht mehr möglich sei. Die Gegenhypothese ist, dass Weinen „nichts als undifferenziertes Rauschen“ sei. Bis heute liegt kein anerkannter Beweis für die eine oder andere Hypothese vor. Es gibt lediglich starke Hinweise dafür, dass sich die Plötzlichkeit des Eintretens des Grundes in den akustischen Eigenschaften bemerkbar macht. Ein plötzliches Ereignis, wie ein Nadelstich oder ein lautes Geräusch, führen auch zu einem plötzlich beginnenden Weinen. Ein langsam eintretendes Ereignis, wie ein langsam zunehmender Schmerz oder Hunger führen auch zu einem langsam eintretenden Weinen. Da nach Kenntnis des Autors bis heute keine wissenschaftlich belastbarer Beweis vorgelegt wurde, wird empfohlen, den Grund aus dem Kontext abzuleiten.[43, S. 9 - 13, 17 - 19]

Die Zweite Frage nach der Ableitung der Stärke des Unwohlseins aus den akustischen Eigenschaften des Weinens wird in der Fachliteratur unter dem Begriff *Cry as a graded Signal* subsumiert. Je „stärker“ das Weinen, desto höher sei das Unwohlsein (*Level of Distress (LoD)*) des Säuglings. Tatsächlich bemessen wird dabei der von dem Beobachter vermutete Grad des Unwohlseins des Babys, und nicht der tatsächliche Grad, da dieser ohne die Möglichkeit der direkten Befragung des Babys nie mit absoluter Sicherheit bestimmt werden kann. Ein hohes Unwohlsein hat vor allem eine schnelle Reaktion der Aufsichtspersonen zur Beruhigung des Babys zur Folge, womit dem Weinen eine Art Alarmfunktion zukommt. Es gibt starke Hinweise darauf, dass das Level of Distress anhand objektiv messbarer Eigenschaften des Audiosignals bestimmt werden kann. So herrscht beispielsweise weitestgehend Einigung darüber, dass ein „lang“ anhaltendes Wein auf einen hohen Level of Distress hinweist. Insofern aus dem Kontext des Weinens Schmerz als die wahrscheinlichste Ursache eingegrenzt werden kann, kann aus einem hohen Level of Distress ein hoher Schmerz abgeleitet werden. [43, S. 13 - 17] [42] Es herrscht wiederum keine Einigung darüber, welche akustischen Eigenschaften im Detail ein hohes Level of Distress anzeigen. Carlo V Bellieni et al. [5] haben festgestellt, dass bei sehr hohem Schmerz in Bezug auf die DAN-Scale (siehe Tabelle 2.2) die Tonhöhe steigt. Qiaobing Xie et al. [42] haben festgestellt, dass häufiges und dysphoniertes Schreien auf einen hohen Level of Distress hinweist.

2.2 Signalverarbeitung

In Kapitel 2.1 wurde erläutert, wie Weinen von Neugeborenen mit Hilfe subjektiv behafteter Begriffe eingeschätzt werden kann. Möchte man das Weinen objektiv beschreiben und messbar machen, so verwendet man die Methoden der digitalen Signalverarbeitung. An dieser Stelle wird eine Einführung in die wichtigsten Themen, die im Zusammenhang mit der Audiosignalverarbeitung größere Bedeutung haben. Es wird ein grundlegendes Verständnis der Signalverarbeitung vorausgesetzt, da aus Platzgründen keine für Neulinge geeignete Einführung in das Themengebiet gewährleisten werden kann. Falls dieses Wissen nicht vorhanden ist, wird zur Einarbeitung das Buch „The Scientist and Engineer's Guide to Digital Signal Processing“ von Steven W. Smith empfohlen.[48], welches kostenlos als E-Book bereitgestellt wird.

2.2.1 Grundlegende Definitionen

In dieser Arbeit sind nur *digitale Signale* von Bedeutung. Ein digitales Signal $x[\,]$ ist nach Formel 2.1 ein beliebige Zahlenfolge mit diskretem Definitionsbereich. Dem Definitionsbereich kommt die Bedeutung *Zeit* zu.[48, S. 11-12] In dieser Arbeit gilt die Konvention, dass mit $x[\ ]$ das gesamte Signal gemeint ist und mit $x[n]$ *ein* Wert des Signals zum Zeitpunkt/Index n. Ein Wert $x[n]$ wird auch als *Sample* bezeichnet. Die Samplingfrequenz des digitalen Signals wird mit f_s bezeichnet.

$$x[\,] := \quad \forall n \in \mathbb{Z} : \ x[n] = s \tag{2.1}$$

Der Definitionsbereich eines Signals erstreckt sich implizit immer von negativer bis positiver Unendlichkeit. Das heißt nicht, dass alle Samples des Signals auch Informationen enthalten müssen. Der *Support* ist das kleinst mögliche Zeitintervall, der alle Samples enthält, die nicht den Wert 0 haben, wie Formel 2.2 definiert. Wird also auf ein Sample zugegriffen, das außerhalb des Supportes liegt, hat dieses Sample den Wert 0 (bezeichnet als „0-Sample“)[36, S. 24]

$$\begin{aligned} \mathrm{Sup}(x[\,]) = [sup_s, sup_e] \quad , sup_s, sup_e \in \mathbb{Z} \\ , x[sup_s] \neq 0 \ \wedge \ x[sup_e] \neq 0 \ \wedge \ \forall n \ \notin [sup_s, sup_e] : x[n] = 0 \end{aligned} \tag{2.2}$$

Die *Dauer* eines Signals ist die Länge des Supportes nach Formel 2.3. In dieser Arbeit herrscht die Konvention, dass die Länge des Signals kurz mit der Variable N abgekürzt wird. Wenn nicht anders definiert, erstreckt sich der Support eines Signals von $0, \ldots, N-1$.[36, S. 24]

$$\mathrm{Length}(x[\,]) = sup_e - sup_s + 1 = N \tag{2.3}$$

2.2.2 Statistische Merkmale

Im folgenden wird ein Überblick über häufig berechnete Merkmale von Signalen gegeben. Abbildung 2.1 visualisiert die Erläuterungen.

1. Der **Maximalwert / Minimalwert** beschreibt den höchsten / niedrigsten in $x[\ ]$ enthaltenen Wert nach den Formel 2.4.

$$\begin{aligned} \max(x[\,]) &= \max_{n \in \mathrm{Sup}(x[\,])} \{\ x[n]\ \} \\ \min(x[\,]) &= \min_{n \in \mathrm{Sup}(x[\,])} \{\ x[n]\ \} \end{aligned} \tag{2.4}$$

2. Der **Durchschnittswert / Average Value** beschreibt den durchschnittlichen Wert aller Samples von $x[\,]$ nach Formel 2.5. Dieser Durchschnittswert wird über ein beliebiges Intervall $[n_1, n_2]$ berechnet.

$$\mathrm{AVG}(x[\,]) = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x[n] \tag{2.5}$$

3. Der **Mean Squared Value** (*MSV*) beschreibt den quadrierten Durchschnittswert über eine bestimmtes Intervall nach Formel 2.6. Er wird auch als *durchschnittliche Energie* oder *average Power* bezeichnet.

$$\mathrm{MSV}(x[\,]) = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x[n]^2 \tag{2.6}$$

4. Das **Root Mean Square** (*RMS*) wird definiert als die Wurzel des Mean Squared Value nach Formel2.7. Der RMS kann im Vergleich zum MSV besser ins Verhältnis zu den Werten des Signals gesetzt werden kann. Er wird im Deutschen auch als **Effektivwert** oder **Durchschnittsleistung** bezeichnet. Da die deutschen Begriffe in einigen Quellen jedoch auch für den MSV verwendet werden, wird in dieser Arbeit nur mit den englischen Begriffen gearbeitet.

$$\mathrm{RMS}(x[\,]) = \sqrt{\frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x[n]^2} \tag{2.7}$$

5. Die **Energie** / **Energy** eines Signals wird nach Formel 2.8 definiert. Sie entspricht dem MSV-Wert multipliziert mit der Länge des Intervalls. [36, S. 27-28]

$$\mathrm{E}(x[\,]) = \sum_{n=n_1}^{n_2} x[n]^2 \tag{2.8}$$

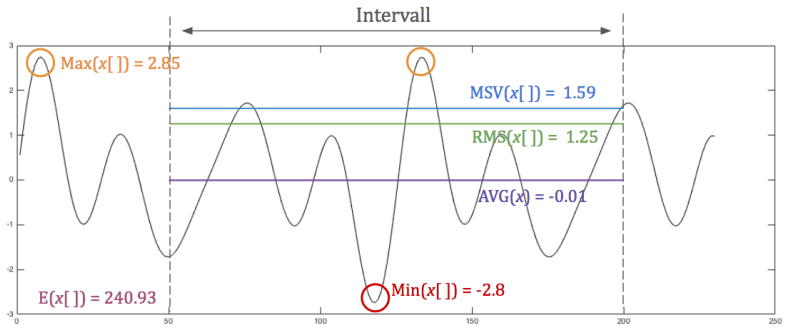

Abbildung 2.1: Statistische Merkmale eines Beispielsignals über dem Intervall [50,200]

2.2.3 Fehlersignale

Angenommen, ein Signal $x[\,]$ wird übertragen, auf dem Übertragungsweg jedoch durch ein anderes Störsignal wie z.B. Rauschen $e[\,]$ überlagert. $e[\,]$ wird in diesem Zusammenhang als das *Fehlersignal* bezeichnet. Das resultierende *Nutzsignal* $x'[\,]$ wird nach Formel 2.9 berechnet.

$$x'[\,] := \quad \forall_{n=n_1}^{n_2} \; : \; x'[n] = x[n] + e[n] \tag{2.9}$$

Eine Möglichkeit der Quantifizierung der Stärke des Rauschens im Vergleich zum Signal ist, den MSV des Eingangssignal ins Verhältnis zum MSV des Fehlersignals zu setzen. Formel 2.10 gibt die Definition.

$$\mathrm{SNR}_{rel}(x[\,], e[\,]) = \frac{MSV(x[\,])}{MSV(e[\,])} \tag{2.10}$$

In der Praxis ist der MSV des Eingangssignals meist sehr viel höher als der des Fehlersignals. Um den Zahlenraum zu begrenzen, wird die Pseudoeinheit dB verwendet. Formel 2.11 definiert den *Signal/Rausch-Abstand* (*SNR*, englisch Signal-to-Noise-Ratio). Ein *niedriger* SNR weist auf ein *starkes* Rauschen hin, und ein *hoher* SNR auf ein *schwaches* Rauschen. Im Zusammenhang mit der Spracherkennung ist der Signal/Rausch-Abstand von Bedeutung, da ein höheres Rauschen die Verarbeitung des Nutzsignals, der Sprache, erschwert.

$$\mathrm{SNR}(x[\,], e[\,]) = 10 \cdot \lg\left(\frac{MSV(x[\,])}{MSV(e[\,])}\right) \text{ dB} \tag{2.11}$$

2.2.4 Kurzzeit-Fourier-Transformation

Das Signal $x[\,]$ befindet sich im *Zeitbereich*, da die unabhängige Variable die Zeit definiert. Gleichung 2.12 definiert die *komplexe diskrete Fouriertransformation*, kurz *DFT*, die das diskrete Signal $x[\,]$ aus dem Zeitbereich in den Frequenzbereich $X[\,]$ transformiert. Das Signal des Frequenzbereiches ist, ebenso wie das Signal des Zeitbereiches, N punkte Lang und hat den Support $0, \ldots, N-1$. Jedes Sample des Frequenzbereiches ist eine komplexe Zahl, deren Realteil $\Re(x[k])$ die Amplitude der entsprechenden Sinuswelle mit der Frequenz $f = k\frac{f_s}{N}$ bezeichnet und deren Imaginärteil $\Im(x[k])$ die Amplitude der entsprechenden Kosinuswelle bezeichnet.[48, S. 149, S. 567 - 571] [1, S. 60]

$$\mathrm{DFT}\{x[\,]\} = X[\,] := \quad \underset{k=0}{\overset{N-1}{\forall}} : \; X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi k \frac{n}{N}} \tag{2.12}$$

Das *Frequenz-Spektrum*, kurz auch nur als *Spektrum* bezeichnet, wird in dieser Arbeit nach Gleichung 2.13 definiert als der Absolutwert des Frequenzbereiches im Bereich $0, \ldots, N/2$.

$$\text{Spektrum} := \quad |X[0]| \;,\; \ldots \;,\; |X[N/2]| \tag{2.13}$$

Abbildung 2.2 visualisiert die Transformation in den Frequenzbereich: In der Abbildung ist oben der Zeitbereich eines 1.8 Sekunden langen Signals zu sehen. Es können klar drei nacheinander gespielte Töne erkannt werden. Der Zeitbereich lässt erkennen, zu welchen Zeitpunkten die Töne beginnen und Enden, aber nicht, welche Frequenzenkomponenten in den Tönen enthalten sind. Das heißt, es kann beispielsweise nicht erkannt werden, ob es sich um hohe oder tiefe Töne handelt. Unten ist das Spektrum abgebildet. Die x-Achse bezeichnet die Frequenz von 0 bis 22 050 Hz und die x-Achse die Amplitude der entsprechenden Frequenz. Beide Achsen werden logarithmiert dargestellt. Das Frequenzspektrum zeigt, welche Frequenzkomponenten im dem Signal enthalten sind. So kann beispielsweise erkannt werden, dass keine Frequenzen unterhalb von 1000 Hz in dem Beispielsignal enthalten sind. Das Spektrum macht jedoch nicht erkennbar, zu welchen Zeitpunkten die Töne beginnen oder enden.

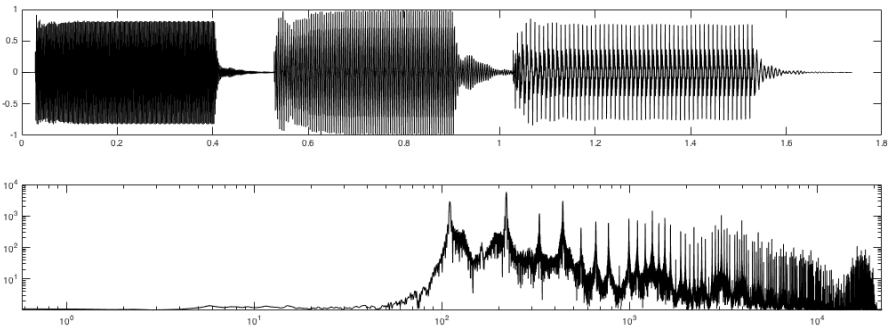

Abbildung 2.2: Ein 1.8-Sekunden langes Signal. Oben: Der Zeitbereich mit drei klar erkennbaren Events. Unten: Das Frequenz-Spectrum des gesamten Signals mit logarithmisierten Achsen.

Es ist wünschenswert, einen Kompromiss aus den Vorteilen beider Bereiche zu finden, in dem man das Spektrum kürzerer Zeitabschnitte des Signals bildet. Dazu wird der Zeitbereich $x[\,]$ in Fenster der Länge M zerlegt. Die zeitliche Differenz zwischen zwei Fenstern wird als *Hoptime* R bezeichnet. Gleichung definiert die Bildung des *Signalfensters* $x_i[\,]$. Die komplette Zerlegung eines Signals in Signalfenster wird als *Windowing* bezeichnet.[47]

$$x_i[\,] := \quad \forall_{n=0}^{M-1} : \; x_m[n] = x[n + i \cdot R] \tag{2.14}$$

Abbildung 2.3 gibt ein Beispiel für die Zerlegung eines Signals $x[\,]$ in die Signalfenster $x_0[\,], \ldots, x_4[\,]$. Die Samplingrate des Signals ist $f_s = 44100$, die Fensterlänge beträgt $M = 22050/f_s = 0.5\,\mathrm{s}$ und die Hoptime $R = M/2 = 0.25\,\mathrm{s}$.

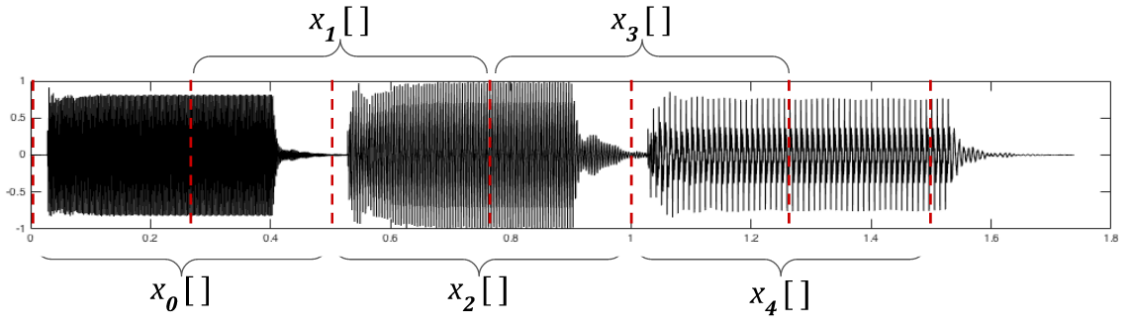

Abbildung 2.3: Windowing: Die Zerlegung eines Signals in kürzere Fenster.

Als Vorbereitungsschritt für die Transformation der Signalfenster in den Frequenzbereich wird nun jedes Fenster mit einer *Fensterfunktion* (engl. *window*) $w[\,]$ multipliziert.[1, S. 69] Gleichung 2.15 definiert eine der am weitesten verbreiteten Fensterfunktionen, das *Hamming-Window*. Der Paramter M gibt die länge des Fensters an. Abbildung 2.4 visualisiert das Hamming-Window. [48, S. 286]

$$w[\,] := \quad \forall_{n=0}^{M-1} : \; w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{M}) \tag{2.15}$$

Die Gleichung 2.16 definiert die *Kurzzeit-Fourier-Transformation* (engl. *Short Time Fourier Transformation*, kurz *STFT*), implementiert mit Hilfe der DFT. Dabei wird das

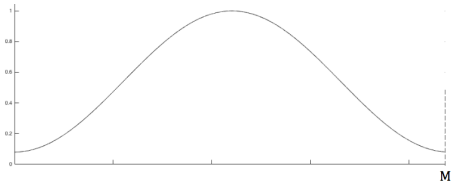

Abbildung 2.4: Das Hamming-Window

Signalfenster $x_i[\,] = x[n + i \cdot R]$ mit der Fensterfunktion $w[\,]$ multipliziert und in das *Frequenz-Fenster* $X_i[\,]$ transformiert.[1, S. 69] [4] Abbildung 2.5 visualisiert die STFT des Beispiels aus Abbildung 2.3.

$$\mathrm{STFT}_i\{x[\,]\} = X_i[\,] := \quad \underset{k=0}{\overset{M-1}{\forall}} : \; X_i[k] = \sum_{n=0}^{M-1} x[n + i \cdot R] \cdot w[n] \cdot e^{-j2\pi k \frac{n}{N}} \tag{2.16}$$

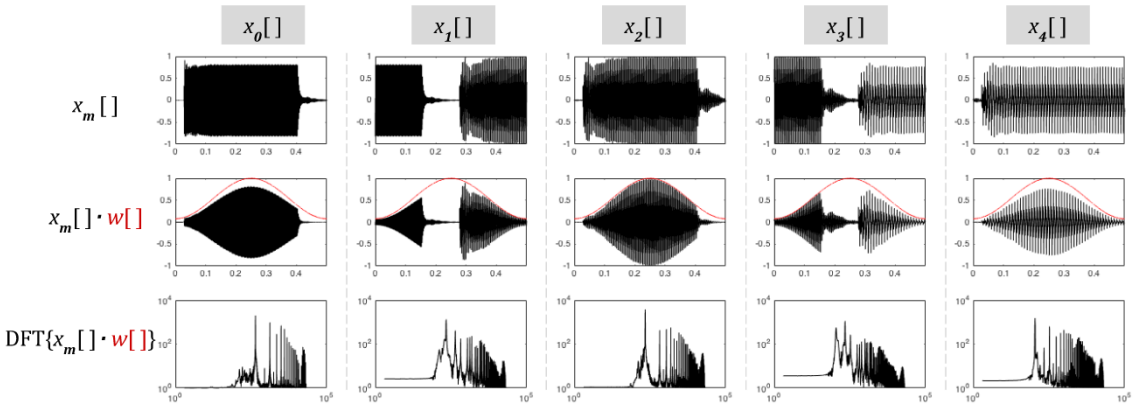

Abbildung 2.5: STFT des Beispiel-Signals aus Abbildung 2.3

2.2.5 Akustische Modellierung der menschlichen Stimme

Der menschliche Sprechapparat wird in die folgenden Komponenten Unterteilt:

Schallproduktion: Die Lunge stößt Luft aus, welche die Stimmbänder passieren. Sind die Stimmbänder leicht gespannt, so wird der Luftstrom periodisch unterbrochen. Die Schwingfrequenz beträgt bei erwachsenen Männern etwa 120 Hz und bei Frauen 220 Hz. Die Frequenz kann während des Sprechens um bis zu einer Oktave variieren. Es wird so ein periodisches, akustisches Signal produziert, bezeichnet als „periodische Quelle“ (engl. „periodic Source“). Sind die Stimmbänder stark gespannt, so entstehen Turbulenzen, die sich akustisch als ein zischendes Geräusch ohne identifizierbare Tonhöhe äußern. Dieses stimmlose Signal wird bezeichnet als „Turbulenzquelle“ (engl. „turbulance Source“)

Klangformung: Das Signal der Stimmlippen passiert den Rachen, Mund- und Nasenraum, welche gemeinsam als „Vokaltrakt“ bezeichnet werden. Das Halszäpfchen bestimmt, ob der Luftstrom in den Mund- oder Nasenraum geleitet wird. Die Stellung der Artikulatoren, bestehend aus dem Kiefer, der Zunge usw. bestimmen die Beeinflussung

des Klanges, der durch die Stimmbänder erzeugt wurde. Diese Beeinflussung wird als Filter angenähert. [17, S. 62] [1, S. 13] Abbildung 2.6 visualisiert diese Komponenten.

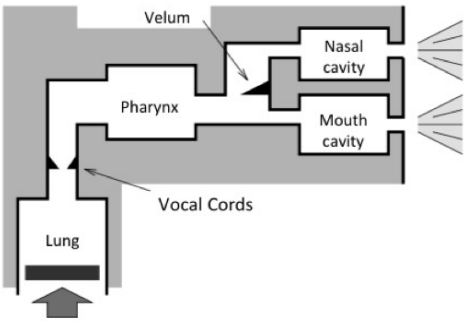

Abbildung 2.6: Schematische Übersicht über die Organe der Spracherzeugung. Lung = Lunge, Vocal Chords = Stimmbänder, Pharynx = Rachen, Velum = Halszäpfchen, Mouth Cavity = Mundraum, Nasal Cavity = Nasenraum [29]

Aus Sicht der Signalverarbeitung wird die menschliche Lautproduktion durch das sogenannte *Source-Filter-Modell* modelliert. Der durch die Stimmbänder erzeugte periodische Ton wird angenähert durch einen Impuls-Zug, welcher durch den Schlund als linearen Filter moduliert wird. Der stimmlose, nicht-periodische Ton wird durch weißes Rauschen angenähert. Der so erzeugte periodische oder nicht-periodische Ton wird als das Eingangssignal $u[\,]$ bezeichnet. Dieses Signal wird daraufhin an den Vokaltrakt weitergeben, welcher als lineares, zeitinvariantes Filter mit der Impulsantwort $v[\,]$ modelliert wird. Diese Impulsantwort ist abhängig von der Konfiguration der Organe des Vokaltraktes. Die Lippen werden als zweites lineares, zeitinvariantes Filter mit der Impulsantwort $r[\,]$ modelliert. $r[\,]$ wird auch als „radiant Model“ bezeichnet. Das tatsächliche Sprachsignal $y[\,]$ entsteht somit durch die Faltung des Signals $u[\,]$ mit den beiden linearen, zeitinvarianten Filtern nach Gleichung 2.17. Gleichung 2.18 definiert den Frequenzbereich des Ausgangssignals $Y[\,]$ durch die Multiplikation der Frequenzbereiche der drei Komponenten. Abbildung 2.7 visualisiert diesen Prozess schematisch. [17, S. 62 - 63] [29]

$$u[\,] * v[\,] * r[\,] = y[\,] \tag{2.17}$$

$$U[\,] \cdot V[\,] \cdot R[\,] = Y[\,] \tag{2.18}$$

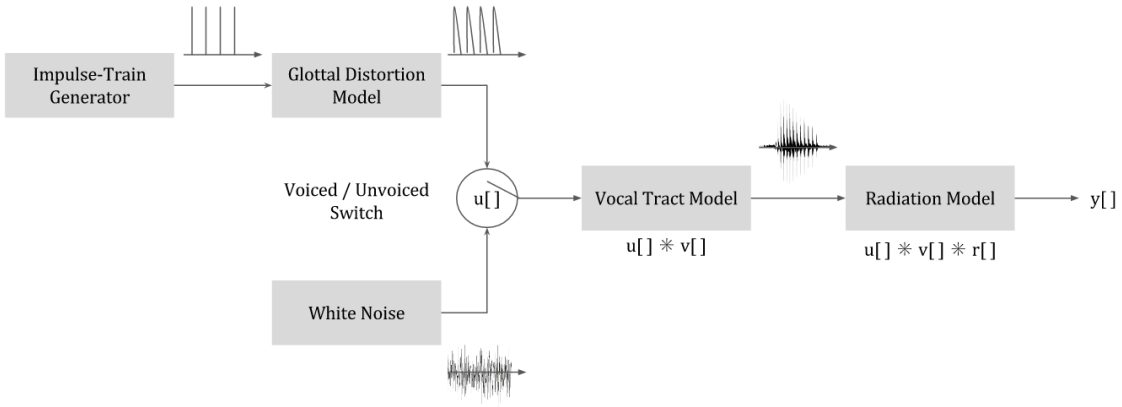

Abbildung 2.7: Schematische Übersicht über das Source-Filter-Model [14, nach Source estimation, S. 17]

Abbildung 2.8 zeigt die Zeitbereiche der stimmhaften und turbulenten Quelle im Vergleich. Wie zu sehen ist, bestimmt der zeitliche Abstand zwischen den Impulsen die Grundfrequenz der Stimme. Dieses Signal $p[\,]$ wird durch den Schlund als Filter $G\{\,\}$ gefiltert, wodurch der Zeitbereich der periodischen Quelle entsteht $G\{p[\,]\} = u_p[\,]$. Darunter ist der Zeitbereich des weißen Rauschen zu sehen. [31, Source]

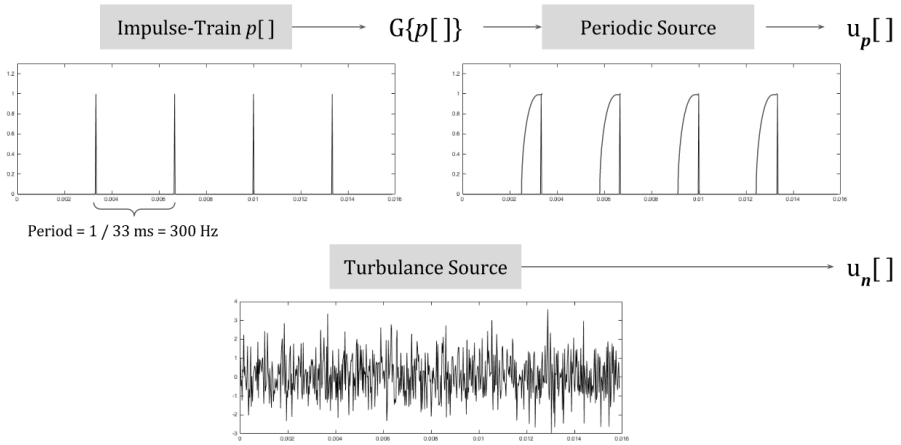

Abbildung 2.8: Zeitbereiche der periodischen und der turbulenten Quelle [31, Source]

Abbildung 2.9 zeigt die Frequenzbereiche der Komponenten des Source-Filter-Modells. Die periodische Quelle ($U[\,]$ links) zeichnet sich im Frequenzbereich durch gleichmäßig verteilte Spitzen aus, die mit steigender Frequenz an Amplitude verlieren. Rechts daneben ist der Frequenzbereich des weißen Rauschen zu sehen, welcher einer Zufallsverteilung entspricht. Die Frequenzantwort des Vokaltraktes $V[\,]$ zeichnet sich durch Resonanzfrequenzen aus, von denen in diesem Beispiel vier erkennbar sind. Die Übertragungsfunktion der Lippen $R[\,]$ wird als Hochpassfilter angenähert. Das Ausgangssignal $Y[\,] = U[\,] \cdot V[\,] \cdot R[\,]$ zeigt den Einfluss der Filter auf das jeweilige Eingangssignal.[14, Source estimation], [31, Vocal Tract Resonance]

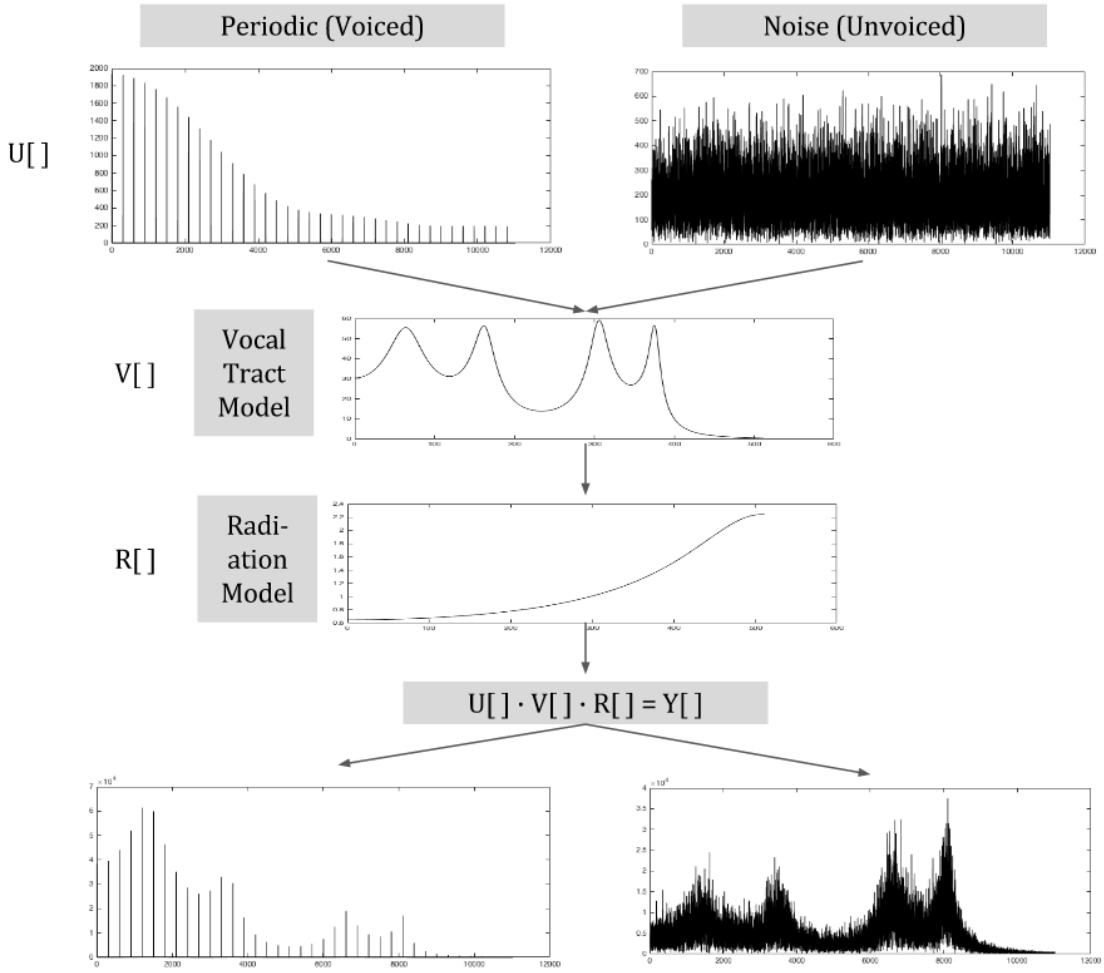

Abbildung 2.9: Betrachtung der Frequenzbereiche des Source-Filter-Model (nach: [14, Source Estimation, S. 3])

Abbildung 2.10 zeigt schematisch das Spektrum eines stimmhaften Sprachsignals. Sowohl die Grundfrequenz als auch die harmonischen Obertonwellen sind rein visuell als „viele, kurze Signalspitzen“ im Spektrum erkennbar. Der kleinste gemeinsame Teiler der Frequenzen dieser Signalspitzen entspricht der Grundfrequenz f_0 des Stimmsignals, in diesem Beispiel 250.7 Hz. Die Grundfrequenz ist ebenfalls an der Signalspitze mit der tiefsten Frequenz ablesbar. Die harmonischen Obertöne entsprechen der doppelten, dreifachen, ... Frequenz dieser Grundfrequenz, das heißt $2 \cdot f_0, 3 \cdot f_0, \ldots$ und werden bezeichnet mit $H_1, H_2, \ldots$. Die Grundfrequenz ist *nicht zwingend* die Spitze der höchsten Amplitude. Durch den Einfluss des Vokaltraktes als Filter können harmonische Oberwellen eine höhere Amplitude als die Grundfrequenz erhalten. Auf Basis des Spektrums lässt sich somit rein visuell ein stimmhaften Signal von einem nicht stimmhaften (Rausch-)Signal unterscheiden, in dem das Spektrum nach dem Vorhandensein dieser regelmäßigen Signalspitzen überprüft wird (vergleiche mit Abbildung 2.9).[1, S. 52 - 53]

Abbildung 2.11 verdeutlicht, wie der als lineares, zeitinvariantes Filter modellierte Vokaltrakt durch Formanten bestimmt wird. Diese Formanten spielen vor allem bei der Beschreibung von Vokalen eine Rolle. Formanten sind lokale Maxima im Spektrum der Transferfunktion, die dadurch erzeugt werden, dass der Vokaltrakt Resonanzen erzeugt. Die

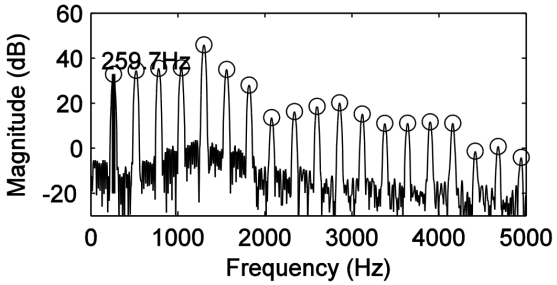

Abbildung 2.10: Grundfrequenz und harmonische Obertöne eines periodischen Sprachsignals.

Formanten werden von links nach rechts durchnummeriert, von $F_1, \ldots, F_n$. Jeder Formant wird durch seine Mittenfrequenz, seine Bandbreite und seine Amplitude beschrieben. Das wichtigste Merkmal ist jedoch die Mittenfrequenz, da sie vom menschlichen Gehör am stärksten zur Identifikation und Unterscheidung der Vokale genutzt wird. Mit steigender Frequenz nimmt die Amplitude der Formanten ab, der dominanteste Formant ist somit immer der erste. Daher werden meist nur die ersten 2 oder 3 Formanten zur Beschreibung eines Vokals angegeben, auch, wenn theoretisch weitaus mehr vom Vokaltrakt erzeugt werden. Für verschiedene Sprachen sind allerlei Tabellen zu finden, welche die Formantenfrequenzen der Vokale auflisten.[1, S. 19]

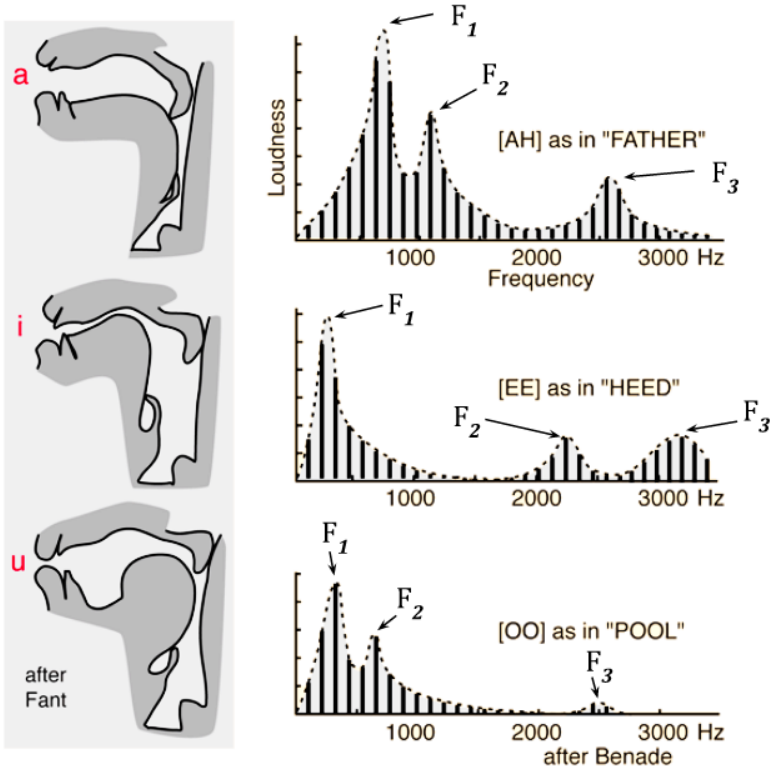

Abbildung 2.11: Formanten im Sprach-Signal (nach: [2])

Beim Sprechen befinden sich sowohl das Signal der Stimmbänder als auch das Filter des Vokaltraktes und der Lippen in ständiger Veränderung. Ein stimmhaftes Sprachsignal

gilt nur über kurze Zeitbereiche weniger Millisekunden als periodisch, und selbst in diesen kurzen Zeitbereichen ist die Stimme nicht perfekt, sondern nur annähernd periodisch. Da die Informationen der Sprache vor allem im Frequenzbereich codiert sind, wird die in Kapitel 2.2.4 vorgestellte Kurzzeit-Fourier-Transformation zur Analyse von Sprache eingesetzt. Die Visualisierung der STFT wird als *Spektogramm* bezeichnet. Dabei werden auf der x-Achse die Zeitpunkte der Fenster und auf der y-Achse die Frequenz dargestellt. Die Frequenzfenster werden „auf die Seite gelegt“, damit ihr zeitlicher Verlauf übersichtlich betrachtet werden kann. Die Amplitude der entsprechenden Frequenzen wird farblich oder durch Helligkeiten codiert, abhängig von der konkreten Implementierung des Spektogramms. Je länger das Zeitfenster der STFT, desto höher ist die Auflösung bezüglich des Frequenzbereiches und desto niedriger die Auflösung bezüglich der Zeitbereiches. Je kürzer die Zeitfenster der STFT, desto höher ist die Auflösung bezüglich des Zeitbereiches, und desto niedriger die Auflösung des Frequenzbereiches.[1, S. 45 - 50] [31, Acoustic Representations of Speech].

Abbildung 2.12 zeigt ein Beispiel für zwei Spektogramme mit unterschiedlichen Fensterlängen der STFT, angewandt auf einer 9 Sekunden langen Aufnahme eines weinenden Babys. Es ist zu erkennen, wie bei der geringeren Fensterlänge der zeitliche Verlauf besser erkennbar, jedoch die einzelnen harmonischen Obertöne weniger gut voneinander unterscheidbar sind. Bei der längeren Fensterlänge sind die Formanten leichter zu unterscheiden, der Beginn und das Ende der Lautäußerungen jedoch schwerer zu lokalisieren.

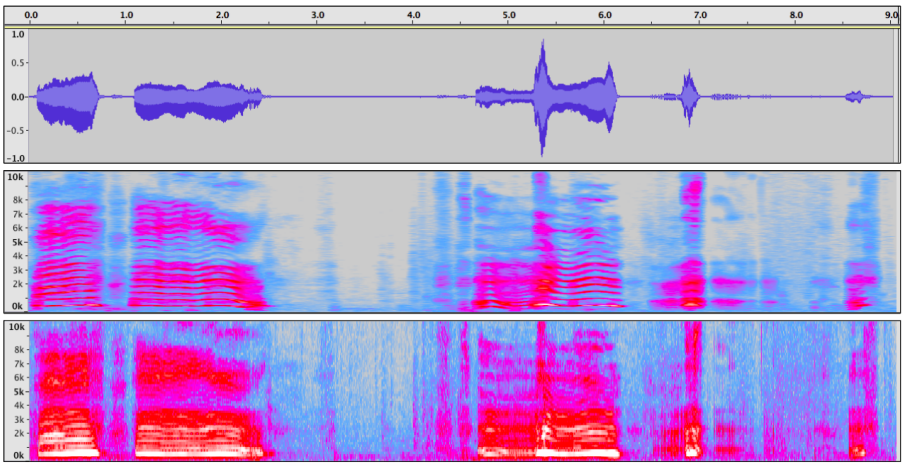

Abbildung 2.12: Spektogramm einer Audioaufnahme eines Babys. Rot $\hat{=}$ hohen Amplituden, Blau $\hat{=}$ niedrigen Amplituden. Oben: Zeitbereich. Mitte: Spektogramm mit einer Fensterlänge von 185 ms(8192-Sample DFT). Unten: Spektogramm mit einer Fensterlänge von 5 ms(265-Sample DFT)

2.3 Schreiforschung

Das Wissenschaftsgebiet, welches sich mit der Analyse und Interpretation von Lautäußerungen Neugeborener auseinandersetzt, wird als „Schreiforschung“ bezeichnet. Die bis heute wohl prominenteste Forschungsgruppe dieses Wissenschaftsgebietes ist die in Kapitel 2.1.2 erwähnte „Scandinavian Cry-Group“[26], welche zwischen 1960 und 1990 die Laute von Babys systematisch erforscht haben. Das wichtigste Werkzeug zur Analyse der Lautäußerungen war das Spektogramm, welches damals auf analogen Technologien basierte. Das Ziel der frühen Schreiforschung war es, mit Hilfe des Spektogramms Muster zur Unterscheidung eines

abnormalem Weinen von einem normalen Weinen zu finden, um beispielsweise Krankheiten erkennen zu können.[43, S. 142]

Teil der Scandinavian Cry-Group waren H. Golub und M. Corwin, die in der Veröffentlichung „A Physioacoustic Model of the Infant Cry“[17] ein Vokabular zur Beschreibung typischer, im Spektogramm erkennbarer Muster festgelegt haben. Da das Vokabular bis heute Einsatz findet, wird an dieser Stelle eine Übersicht über die wichtigsten Begriffe gegeben. Weiterhin werden Begriffe eingeführt, die von Zeskind et al. in „Rythmic organization of the Sound of Infant Cry “ veröffentlicht wurden.[40]

2.3.1 Phyisio-Akustische Modellierung des Weinens

Das Weinen von Babys lässt sich im allgemeinen als das „rhythmische Wiederholen eines beim Ausatmen erzeugen Geräusches, einer kurzen Pause, einem Einatmungsgeräusch, einer zweiten Pause, und dem erneuten Beginn des Ausatmungsgeräusches“beschreiben. [52].

Die folgenden Begriffe werden in Abbildung 2.13 veranschaulicht.

- **Expiration (Ausatmung):** Der Klang, der bei einem einzelnen, ununterbrochenen Ausatmen mit Aktivierung der Stimmbänder durch das Baby erzeugt wird. [40]. Der von Golub et al. [17, S. 61] verwendete Begriff **Cry-Unit** wird in dieser Arbeit synonym verwendet. Umgangssprachlich ist handelt es sich um einen einzelnen, ununterbrochenen *Schrei*.
- **Inspiration (Einatmung):** Der Klang, der beim Einatmen durch das Baby erzeugt wird.
- **Burst:** Die Einheit einer Ausatmung und der darauf folgenden Einatmung. Das heisst, dass die zeitliche Dauer eines Bursts sowohl die Ausatmung, die Einatmung als auch die beiden Pausen zwischen diesen Geräuschen umfasst.[40][1]
- **Cry:** Die gesamte klangliche Antwort zu einem spezifischen Stimulus. Eine Gruppe mehrerer Cry-Units.[17, S. 61] In dieser Arbeit wird ein *Cry* auch als **Cry-Segment** bezeichnet, um Verwechslungen zu vermeiden.

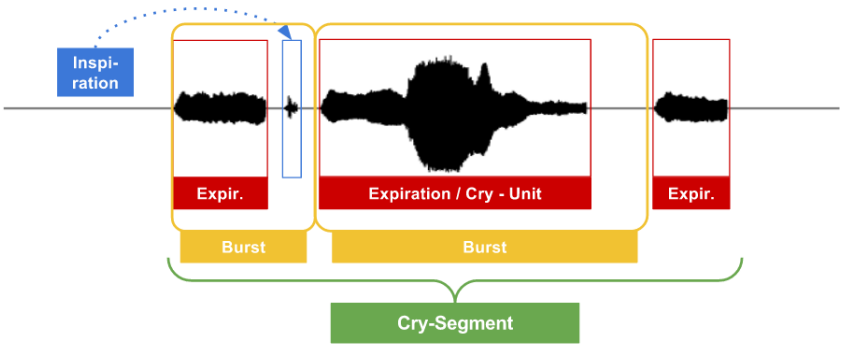

Abbildung 2.13: Veranschaulichung des Grundvokabulars

[1] Praktisch ergibt sich das Problem, dass vor allem bei stärkerem Hintergrundrauschen die Einatmung häufig weder hörbar noch auf dem Spektrogramm erkennbar ist. Daher wird die Zeitdauer eines Bursts von Beginn einer Ausatmung bis zum Beginn der darauf folgenden Ausatmung definiert und somit allein von den Ausatmungsgeräuschen auf die Bursts geschlossen. Implizit wird somit eine Einatmung zwischen zwei Ausatmungen angenommen.

Cry-Units werden von H. Golub und M. Corwin in eine der drei folgenden Kategorien eingeordnet, bezeichnet als *Cry-Types*: [17, S. 61 - 62]

- **Phonation** beschreibt eine Cry-Unit mit einer „vollen Vibration der Stimmbänder“ und einer Grundfrequenz zwischen 250 und 700 Hz. Entspricht umgangssprachlich einem Weinen mit einem „klaren, hörbaren Ton“.
- **Hyper-Phonation** beschreibt eine Cry-Unit mit einer „falsetto-artigen Vibration der Stimmbänder“ mit einer Grundfrequenz zwischen 1000 und 2000 Hz. Entspricht umgangssprachlich einem Weinen mit einem „sehr hohen, aber klar hörbaren Ton“.
- **Dysphonation** beschreibt eine Cry-Unit ohne klar feststellbare Tonhöhe, produziert durch Turbulenzen an den Stimmbändern. Entspricht umgangssprachlich dem „Brüllen oder Krächzen“.

Die folgenden weiteren Eigenschaften können für einzelne Cry-Units extrahiert werden:

- **Duration:** Die zeitliche Dauer der Cry-Unit.
- **Duration of Inspiration:** Die zeitliche Dauer der Pause zwischen zwei Cry-Units.
- **Grundfrequenz:** Für eine Cry-Unit kann die durchschnittliche, die höchste und die niedrigste Grundfrequenz sowie die Varianz festgestellt werden.
- **Frequenz der Formanten:** Wie bei der Grundfrequenz kann der Durchschnitt, das Maximum, Minimum usw. für eine Cry-Unit berechnet werden.
- **Ratio2:** Verhältnis zwischen den Energien der Frequenzen unterhalb von 2000 Hz zu den Frequenzen oberhalb von 2000 Hz
- **Cry-Mode Changes:** Häufigkeit des Wechsels des Cry-Modes innerhalb einer Cry-Unit.
- **Amplitude:** Die Lautstärke der Cry-Unit, gemessen in Dezibel. [27, S. 85] [10, S. 156]

H. Golub und M. Corwin haben weiterhin eine Reihe von Eigenschaften vorgestellt, die das zeitliche Verhalten der Grundfrequenz und der harmonischen Obertöne innerhalb einer Cry-Unit beschreiben. [17, S. 73] Einige dieser Eigenschaften werden in Abbildung 2.14 in einem schematischen Spektogramm dargestellt. Die schwarzen Linien zeigen den Verlauf der Grundfrequenz und der Formanten.

- **Pitch of Shift:** Grundfrequenz nach einem schnellen Anstieg zu Beginn der Cry-Unit
- **Glide:** Kurzes, starkes ansteigen der Grundfrequenz
- **Glottal Roll:** Dysphonation, die häufig am Ende einer Cry-Unit nach einem Abfall der Grundfrequenz beobachtet wird.
- **Vibrato:** Mehr als vier starke Schwankungen der Grundfrequenz innerhalb einer Cry-Unit.
- **Melody-Type:** einer Cry-Unit. Meist: fallend, steigend/fallend, steigend, fallend/-steigend, flach.
- **Continuity:** Verhältnis zwischen stimmhaften und nicht-stimmhaften Bereichen der Cry-Unit

- **Double Harmonic Break:** Das Aufkommen einer zweiten Serie von harmonischen Obertönen zwischen den eigentlichen harmonischen Obertönen der Cry-Unit.
- **Biphonation:** Das Aufkommen einer zweiten Grundfrequenz mit eigenen harmonischen Obertönen zusätzlich zu der eigentlichen Grundfrequenz.
- **Noise Concentration:** Starke Energiespitzen zwischen 2000 und 2300 Hz.
- **Furcation:** Plötzliches Aufteilen der Grundfrequenz und harmonischen Obertöne in mehrere, schwächere Obertöne.

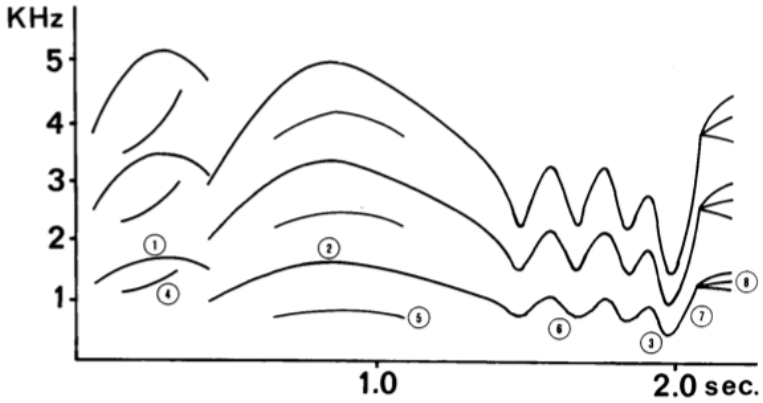

Abbildung 2.14: (1) Pitch of Shift (2) Maximale Grundfrequenz (3) Minimum der Grundfrequenz (4) Biphonation (5) Double Harmonic Break (6) Vibrato (7) Glide (8) Furcation [43, S. 142]

Die folgenden Eigenschaften werden in Bezug auf das gesamte Cry-Segment, oder zumindest auf eine Menge aufeinander folgender Cry-Units berechnet:

- **Cry Latence:** Zeit zwischen Stimulus, wie zum Beispiel einem Nadelstich, und der ersten Cry-Unit.
- **Utterances:** Anzahl der Cry-Units im Segment.
- **Short Utterances:** Anzahl stimmloser Cry-Units im Segment.
- .... und statistische Auswertungen bezüglich aller oben genannten Features, die sich auf eine Cry-Unit beziehen, wie beispielsweise der Durchschnitt aller Tonhöhen, Anzahl des Vorkommens bestimmter Melodiekonturen, Varianz der Länge der Cry-Units etc.[27, S. 85]

Einige Krankheiten wurden in Zusammenhang mit dem vermehrten Vorkommen bestimmter Eigenschaften bei kindlichen Lautäußerungen gebracht. So wurde eine Korrelation zwischen dem Anstieg der durchschnittlichen Grundfrequenz, häufiger Biphonation und geringer Duration in Zusammenhang mit Gehirnschäden beobachtet. Tendenziell niedrige Grundfrequenzen zeigen eine Korrelation mit Trisomie 13, 18 und 21.[27, S. 85]

2.3.2 Diskussion

Bis heute bleibt die Analyse von kindlichen Lautäußerungen weitestgehend unstandartisiert: [43, S. 142]

- Es gibt keine komplette Liste, welche einen Überblick über alle berechenbaren Eigenschaften für Cry-Units oder Segmente gibt. Viele Veröffentlichungen beziehen sich auf die Eigenschaften, die von H. Golub und M. Corwin vorgestellt wurden, und erweitern diese Liste mit eigenen Vorschlägen.
- Es gibt keine Einigung darüber, welche der Eigenschaften die wichtigsten sind. Beispielsweise konzentrierten sich H. Golub und M. Corwin [17] vermehrt auf die Erkennung von Mustern im Melodieverlauf, Zeskind et al. auf zeitliche Eigenschaften.[40]. Die Eigenschaft, die am häufigsten mit Schmerz, Krankheiten und sonstigen Abnormalitäten in Verbindung gebracht wird, ist eine abnormal hohe oder niedrige Tonhöhe. Bei einigen Features, die von H. Golub und M. Corwin verwendet wurden, ist nicht einmal gesichert, ob es sich nicht doch um technische Artefakte der damals verwendeten Analogtechnik handelt. [27, S. 84 - 85]
- Selbst, wenn in verschiedenen Studien die selbe Eigenschaft verwendet wird, wie zum Beispiel die durchschnittliche Tonhöhe, ist nicht standardisiert, wie dieses zu berechnen ist. Mit „durchschnittliche Tonhöhe des Segmentes“ kann gemeint sein: 1.) die durchschnittliche Tonhöhe, errechnet aus den durchschnittlichen Tonhöhen der der Cry-Units. 2.) Die durchschnittliche Tonhöhe aller festgestellten Tonhöhen. 3.) Die durchschnittliche Tonhöhe nur von Ausatmungslauten usw.
- Zusammenhänge, die zwischen bestimmten Eigenschaften des Weinens und bestimmten Krankheitsbildern festgestellt wurden, haben häufig eine hohe Spezifität, aber niedrige Sensitivität. So wurde zum Beispiel festgestellt, dass Kinder, die am plötzlichen Kindstot sterben, fast immer eine Erhöhung der Frequenz des ersten Formanten in Verbindung mit häufigen Cry-Mode-Changes zeigen. Viele Babys, die nicht am plötzlichen Kindstot sterben, zeigen jedoch die selben Merkmale.[27, S. 85]
- H. Golub und M. Corwin behaupten, bereits in den achtziger Jahren ein System zur computergestützten und voll automatisierten Analyse von Cry-Segmenten implementiert zu haben. Das System nimmt 1.) eine Audioaufnahme, gespeichert auf einer Kasette an, 2.) berechnet Formanten, Grundfrequenz und Amplitude gegen die Zeit, 3.) samplt die Grundfrequenz-Kontur, 4.) berechnet insgesamt 88 akkumulierte Features für das gesamte Segment und 5.) zieht Schlussfolgerungen aus den 88 Features, wie zum Beispiel die Diagnose einer bestimmten Krankheit.[17, S. 75 - 76] Abseits der kurzen Erwähnung der Existenz dieser „Mutter aller automatisierten Analysesysteme für das Weinen von Babys“ konnte der Autor dieser Arbeit keine Implementierungsdetails oder sonstige genaueren Ausführungen über das System finden, welche für diese Arbeit von höchstem Interesse gewesen wären.

2.4 Klassifizierung und Regression

Klassifizierung und Regression sind Teilgebiete des Wissenschaftsgebietes des *Überwachten Lernens*, einem Teilgebiet des Wissenschaftsgebietes des *maschinellen Lernens*. Das Ziel beim Überwachten Lernen ist es, einen *Prädiktor*, auch bezeichnet als*Modell*, zu entwerfen, der aus den Eigenschaften einer Instanz dessen Kategorie oder Wert ableiten kann. Im Zusammenhang mit der Schreiforschung könnte eine Instanz eine Baby sein, dessen Eigenschaften 1.) die durchschnittliche Tonhöhe beim Weinen und 2.) die Augenfarbe ist. Der Prädiktor hat nun die Aufgabe, aus diesen beiden Eigenschaften eine Klasse abzuleiten, wie zum Beispiel das Geschlecht des Babys, oder einen Wert, wie beispielsweise das Alter. Das

Lernen basiert dabei aus dem Generalisieren einer Liste von Beispielen, die der Algorithmus zur Verfügung gestellt bekommt. In diesem Zusammenhang wäre dies eine Liste an Babys, bei der für jede Instanz das Geschlecht oder das Alter bereits bekannt ist. Der Algorithmus versucht nun, diese Beispiele soweit zu Verallgemeinern, dass er für neue, bisher unbekannte Babys die Klasse oder den Wert korrekt voraussagen kann.[32, S. 6 - 7]

Eine Instanz x ist ein Vektor $x = (f_1 \in F_1, \ldots, f_n \in F_n)$. F_i wird in diesem Zusammenhang als *Eigenschaft*, *Feature* oder *Attribut* bezeichnet. In Bezug auf das eben genannte Beispiel wäre das erste Feature $F_1 =$ *durschnittliche Tonhöhe* und das zweite Feature $F_2 =$ *Augenfarbe*. Eine Instanz wäre in diesem Fall ein Tupel mit zwei beliebigen Werten dieser Attribute, wie zum Beispiel $x = (300\,\text{Hz}, \text{blau})$). Features, die einen kontinuierliche Wertebereich mit einem quantitativem Charakter haben, wie zum Beispiel das Gewicht, werden als *kontinuierliche* Features bezeichnet. Features, die einen diskreten Wertebereich mit einem qualitativem Charakter haben, wie zum Beispiel die Augenfarbe, werden als *diskrete* Features bezeichnet. Die Menge aller möglichen Kombination der Features $F_1 \times, \ldots, F_n$ wird als *Feature-Raum* bezeichnet. Der Trainings-Datensatz $D_{trainig}$ besteht aus einer Liste an Instanzen, wobei für jede Instanz die Kategorie oder der Wert, gemeinsam Bezeichnet als *Output* oder *Target* $y \in Y$, bekannt ist. Ein Tupel aus einer Instanz zu einem Output wird als *Example* $e = (x, y)$ bezeichnet. Y bezeichnet die Menge aller möglichen Outputs des Problems. Das heißt, $D_{training} = \big(\ (x_1, y_1), \ldots, (x_N, y_N)\ \big)$. Der Prädiktor P ist nun eine Funktion, die von einer Instanz auf den Output abbildet, also $P : X \mapsto Y$. Die Fehlerfunktion E berechnet, wie häufig sich der Prädiktor bei der Bestimmung der Targets eines Test-Datensatzes D_{test} irrt. Der Test- und der Trainings-Datensatz können die selben Instanzen, teilweise die selben oder gar keine gemeinsamen Instanzen beinhalten.[32, S. 6 - 7, 18 - 19] [7, S. 8 - 9]

Bei der **Klassifizierung** wird eine Target als *Klasse* bezeichnet. Die Menge aller möglichen Klassen eines bestimmten Problems $Y = \{y_1, \ldots, y_n\}$ ist dabei diskret und hat einen *qualitativen* Charakter. Das heißt, dass keine Klasse „besser“ oder „höher“ ist als eine andere. Ein Beispiel für ein Klassifizierungsproblem wäre die also die Ableitung des Geschlechtes für eine Instanz, also $Y = \{m, w\}$. Der Prädiktor wird in diesem Fall als Klassifikator C bezeichnet.[2] [11, S. 28, 127]

Bei der *Regression* ist die Menge der möglichen Targets eines bestimmten Problems *kontinuierlich* und hat einen „quantitativen Charakter. Das heißt, es kann eine interne Ordnung in der Menge der Outputs festgelegt werden. Ein Beispiel für ein Regressions-Problem wäre die also die Ableitung des Alters des Babys, also $Y = \{0, \ldots, 3\}$. Der Prädiktor wird in diesem Fall auch als *Regressor* R bezeichnet.[7, S. 24] [32, S. 8] [11, S. 28]

Es gibt eine Vielzahl an Algorithmen zum Finden des Klassifikators oder Regressors. Welcher Algorithmus der „beste“ ist, das heißt für einen Test-Datensatz eine möglichst hohe *Genauigkeit* oder einen möglichst geringen *Klassifikationsfehler* erzeugt, ist abhängig von der konkreten Problemstellung. Auf die Bestimmung der Genauigkeit wird weiter in Kapitel 2.4.2 eingegangen. Ein Algorithmus, der in dieser Arbeit zur Klassifizierung eingesetzt wird, ist der *ID3*- und der *C4.5*-Algorithmus, welcher genauer in Kapitel 2.4.1 beschrieben wird.

[2] In einigen Quellen werden die Begriffe *Klassifizierung* und *Klassifikation* inkonsistent verwendet. *Klassifizierung* ist ein Prozess, dessen Ergebnis die *Klassifikation* ist. Daher wird von einem *Klassifizierungs-Algorithmus* gesprochen, da sich der Algorithmus auf den Prozess des Klassifizierens konzentriert, aber vom *Klassifikationsfehler*, da der Fehler des Ergebnisses der Klassifizierung bestimmt wird. In einigen Fällen ist es unerheblich, ob das Ergebnis oder der Prozess im Fokus des Diskussionsgegenstandes steht. Dann können *Klassifikation* und *Klassifizierung* synonym gebraucht werden.

2.4.1 ID3 und C4.5

Der ID3-Algorithmus zählt zu den sogenannten Entscheidungsbäumen, da der durch den Algorithmus entworfene Klassifikator die Form eines Entscheidungsbaumes annimmt. Die Vorraussetzung ist, dass alle Features diskret und nicht kontinuierlich sind. Tabelle 2.3 zeigt einen Beispieldatensatz zur Erläuterung des Algorithmus. Die Instanzen sind Babys, die Features die „Häufigkeit des Weinens“ und die „Lautstärke des Weinens“, und die beiden Klassen { Ja, Nein } geben an, ob das Kind an chronischem Schmerz leidet

Tabelle 2.3: Beispieldatensatz D für die Kassfikation mit ID3

x_i	$f_1 \in$ Häufigkeit	$f_2 \in$Lautstärke	$y_i =$ chron. Schmerz
x_1	oft	laut	Ja
x_2	selten	laut	Ja
x_3	normal	leise	Nein
x_4	selten	leise	Nein
x_5	normal	laut	Ja
x_6	oft	leise	Ja

Abbildung 2.15 zeigt einen Klassifikator, den der ID-3 Algorithmus für diesen Datensatz erzeugt. Es handelt sich um einen Entscheidungsbaum. In Jedem Knoten steht ein Feature, welches einen Ast für jeden möglichen Wert dieses Features bildet. In den Blättern stehen die Klassen.[32, S. 134]

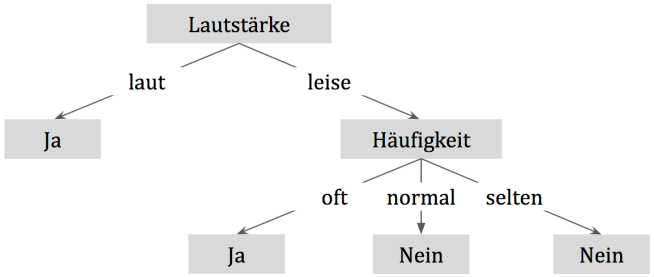

Abbildung 2.15: Entscheidungsbaum, der durch den ID3-Algorithmus für den Datensatz aus Beispiel 2.3 erzeugt wurde.

Der Entscheidungsbaum lässt sich in eine Reihe von `if ... then ...`-Regeln transformieren. Jeder Weg von der Wurzel bis zu einem Blatt ergibt eine Entscheidungsregel, bei der die Feauture-Werte der entsprechenden Kanten konjunktiv Verknüpft werden und die Klasse implizieren. Die Entscheidungsregeln für den Baum aus Abbildung 2.15 sind: [32, S. 134]

- `if` *Tageszeit* = *Tag* `then` *Spaß* = *Ja*
- `if` *Tageszeit* = *Nacht* `and` *Temperatur* = *warm* `then` *Spaß* = *Ja*
- ...

Der Entscheidungsbaum wird beim ID3 Algorithmus nach folgenden Muster erstellt: Die Konstruktion wird Top-Down vollzogen, dass heisst beginnend bei der Wurzel bis zu den Blättern. In jedem Knoten wird ein Feature in alle seine möglichen Attribute aufgespalten.

Um an der Wurzel zu entscheiden, welches Feature zuerst aufgespalten werden soll, wird jedes Feature einem statistischen Test unterzogen, um festzustellen, wie „gut“ dieses Feauture zur Klassifikation der Trainings-Daten beiträgt. Das „beste“ Attribut wird ausgewählt und als Wurzel festgelegt. Nun wird ein Kind für jeden möglichen Wert des Features gebildet. Der Datensatz des Elternknotens wird in disjunkte Teilmengen aufteilt, wobei jedes Kind die Untermenge mit denjenigen Instanzen erhält, die den jeweiligen Feature-Wert besitzen. Daraufhin beginnt für jedes Kind der Prozess des Auswählen des „besten“ Attributes von vorn. Ein Kind wird dann zu einem Blatt, wenn seine Teilmenge an Daten nur noch aus Instanzen einer Klasse besteht und somit kein weiteres Aufteilen notwendig ist.[33, S. 55]

Zur Quantifizierung der Information wird die Entropie nach Formel 2.19 als Hilfsmittel definiert. p_i ist die Wahrscheinlichkeit, dass in einem Datensatz D ein Example mit der Klasse $i \in Y$ angetroffen wird. Die Entropie quantifiziert die *Unreinheit des Datensatzes*. Ein Datensatz, dessen Instanzen alle der selben Klasse angehören, hat die Entropie 0. Ist die *Unreinheit des Datensatzes* hingegen maximal, das heißt, dass der Datensatz exakt gleich viele Instanzen jeder Klasse beinhaltet, ist die Entropie 1. [32, S. 135]

$$H(p) = -\sum_{i \in Y} p_i \cdot \log_2 p_i \tag{2.19}$$

Es ist das Attribut in einem Knoten zu wählen, welches den höchsten *Informatoinsgewinn* gewährleistet, das heißt, zu einer bestmöglichen *Reinheit* in den Kindsknoten der alleinigen Unterteilung des Datensatzes auf Basis dieses Attributs führt. Der Informationsgewinn eines Features F für den Datensatz D wird nach Formel 2.20 definiert. f sind alle möglichen Werte dieses Features. $|D|$ beschreibt die Anzahl an Instanzen des Datensatzes. D_f ist die Untermenge an Instanzen, die für das Feature F den Wert f besitzen.[32, S. 136 - 137]

$$\text{Gain}(D, F) = H(D) - \sum_{f \in F} \frac{|D_f|}{|D|} H(D_f) \tag{2.20}$$

Die Erstellung eines Entscheidungsbaumes mit Hilfe des ID3-Algorithmus wird folgendermaßen als Pseudocode definiert. Der Input des Algorithmus ist der Datensatz D und der Feature-Raum F_{all}, der Output ist der Entscheidungsbaum.[32, S. 139]

ID3(D, F_{all})

- **Wenn** alle Examples $e \in D$ das selbe Label haben:
  - **return** eine Blatt mit diesem Label
- **Sonst: Wenn**
  - **return** ein Blatt mit dem häufigsten Label in dem Datensatz
- **Sonst:**
  - Wähle ein Feature $\hat{F}$ als den nächsten Knoten, dass den Informationsgewinn für den Datensatz D nach Formel 2.20 maximiert.
  - Füge einen Ast für jeden möglichen Wert $f \in \hat{F}$ von dem Knoten hinzu.
  - Für jeden Ast:

- Berechne D_f, in dem $\hat{F}$ von der Liste der Features entfernt wird.
- Rufe **ID3**$(D_f, F_{all}/\hat{F})$ rekursiv auf.

Der ID3-Algorithmus hat folgende **Nachteile**

- Der Algorithmus akzeptiert keine kontinuierlichen Features.[33, S. 72]
- Der Algorithmus neigt zu *Overfitting*. Overfitting bedeutet, dass der Klassfikator C zwar einen möglichst geringen Klassifikationsfehler in Bezug auf den Trainings-Datensatz erzeugt, es jedoch einen anderen Klassifikator C' gibt, welcher für den Trainings-Datensatz einen höheren Fehler erzeugt, jedoch einen geringeren Fehler als C in Bezug auf *alle möglichen Instanzen dieses Typs* erzeugt. Anders formuliert bedeutet Overfitting, dass der Klassifikator den Trainings-Datensatz „auswendig gelernt hat“ und nicht genügend generalisiert, um auf im Training nicht enthaltene Instanzen angewandt werden zu können. Overfitting im Zusammenhang mit dem ID-3 Algorithmus wird durch *Rauschen im Trainings-Datensatz* bedingt.
- Der Algorithmus bevorzugt greedy Attribute, die zum Zeitpunkt der Berechnung den höchsten Informationsgewinn gewährleisten. Dabei besteht die Gefahr, dass der Algorithmus in ein lokales Maximum läuft.[33, S. 66 - 70]

Der *C4.5*-Algorithmus erweitert den *ID3*, um dessen Nachteile auszumerzen, das heißt die Möglichkeit der Verwendung kontinuierlicher Attribute sowie Lösungsansätze für das Overfitting.

Bei einem kontinuierlichen Attribut wird beim *C4.5*-Algorithmus im Gegensatz zu einem diskreten Attribut *nicht* ein Ast für jeden möglichen Wert gebildet, sondern genau zwei Äste. Es wird also ein Grenzwert für das Feature festgelegt, bei dessen Unterschreitung der linke, und bei dessen Überschreitung der rechte Ast gewählt wird (oder, je nach Implementiert, umgedreht). Das Vorgehen zum finden eines solchen Grenzwertes ist wie folgt:

1. Ordne alle Examples nach ihrem jeweiligen Wertes des kontinuierlichen Feature F_c, für das der Grenzwert gesucht wird.
2. Identifiziere benachbarter Examples mit unterschiedlichen Klassen. Die Feature-Werte dieser Examples sind mögliche Kandidaten für einen Grenzwert.
3. Berechne den Informationsgewinn bei Setzung des Grenzwertes auf jeden gefundenen Kandidaten.
4. Wähle denjenigen Grenzwert, der den höchsten Informationsgewinn bringt. [33, S. 73]

Abbildung 2.16 visualisiert einen Knoten mit einer kontinuierlichen Variable F_c, der nach einem Grenzwert in zwei Äste aufgespalten wird.

Das als Overfitting beschriebene Problem lässt sich vermeiden, in dem die Tiefe des Entscheidungsbaumes reduziert wird. Diese Begrenzung wird als *Beschneiden* oder *Pruning* bezeichnet. Es gibt grundlegend zwei verschiedene Ansätze:

Pre-Pruning: Ab der Überschreitung einer bestimmten Tiefe wird der Algorithmus frühzeitig gestoppt und ein Knoten, welcher die maximale Tiefe überschreitet, zwangsweise zu einem Blatt umgewandelt.

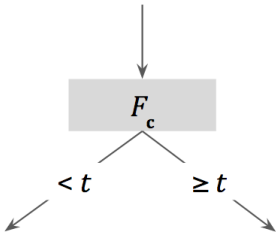

Abbildung 2.16: Aufspaltung einer kontinuierlichen Variable im Entscheidungsbaum

Post-Pruning: Zuerst wird der komplette Entscheidungsbaum aufgebaut und Overfitting zugelassen. Im Nachhinein wird der Entscheidungsbaum in seiner Tiefe reduziert. Eines der am weitesten verbreiteten Post-Pruning-Algorithmen ist das sogenannte *Reduced Error Pruning*. Dabei wird ein Knoten des Entscheidungsbaumes zu einem Blatt umgewandelt und diesem Blatt das Label zugewiesen, welches in seinem Sub-Baum am häufigsten vorkommt. Daraufhin wird der originale Entscheidungsbaum und sowie der beschnittene Entscheidungsbaum verwendet, um den Test-Datensatz zu klassifizieren. Ist der Klassifizierungsfehler des beschnittenen Baumes nicht schlechter als der des originalen Baumes, wird das Pruning übernommen. Dieses Vorgehen wird für jeden Knoten des Entscheidungsbaumes angewandt. [33, S. 68 - 70]

2.4.2 Gütemaße binärer Klassifikatoren

Ein binärer Klassifikation ist eine, bei dem es nur zwei Klassen gibt, das heißt $|Y| = 2$. Applikationsabhängig werden die beiden Klassen beispielsweise als *Positive* und *Negative*, 1 und 0 oder *True* und *False* bezeichnet. Wird bei einer Klassifizierung ein tatsächliches Positive korrekt als Positive vorhergesagt wird, spricht man von einem *True Positive* [TP]. Wird hingegen ein tatsächliches Positive fälschlicherweise als Negative vorhergesagt, spricht man von einem *False Negative* [FN]. Bei der Klassifizierung von Negatives spricht man dementsprechend von *True Negatives* [TN] und *False Positives* [FP]. Die *Confusion Matrix* in Abbildung 2.17 gibt eine Übersicht über die vier möglichen Klassifikations-Ergebnisse. [25, S. 213 - 214]

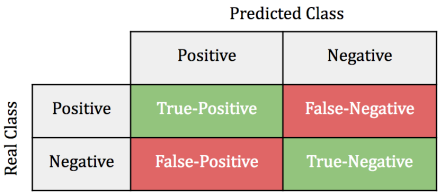

Abbildung 2.17: Confusion-Matrix (nach: [25, S. 214])

Die insgesamte Güte einer Klassifikation wird durch die *Genauigkeit* (engl. *Accuracy*) nach Formel 2.21 bestimmt. Eine Genauigkeit von 100% bedeutet, dass *alle* Instanzen richtig klassifiziert wurden, eine Genauigkeit von 50% bedeutet, dass die Hälfte aller Instanzen richtig klassifiziert wurden. Je höher die Genauigkeit, desto geringer der Klassifikationsfehler. [25, S. 214]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.21}$$

Mit Hilfe der Genauigkeit lässt sich die insgesamte Performance des Klassifikators messen. Der Wert allein gibt jedoch keinen Aufschluss darüber, ob der Klassifikator eher eine Tendenz zur falschen Klassifizierung von Positives oder Negatives hat. Bei einer Datenbank mit der selben Anzahl an Positives und Negatives kann eine Genauigkeit von 50% beispielsweise dadurch entstehen, dass *alle* Instanzen als Positives markiert werden. Das heißt, dass alle Positives richtigerweise als Positives, aber alle Negatives fälschlicherweise ebenfalls als Positives klassifiziert werden. Im Umgedrehten Fall ergibt die Klassifizierung aller Instanzen als Negatives ebenfalls eine Genauigkeit von 50%. In einem dritten Fall irrt sich die Klassifikator gleich oft bei der Einordnung der Negatives und Positives.

Die Maße *Sensitivität* (engl. *Sensitivity*) und *Spezifität* (engl. *Specificity*) geben Aufschluss über die Performance des Klassifikators bei der Prädiktion der Positives und Negatives. Die Sensitivität, auch bezeichnet als *True-Positive-Rate*, bemisst den Anteil tatsächlicher Positives, die auch als solche erkannt wurden, nach Formel 2.22. Eine Sensitivität von 100% bedeutet, dass alle in der Datenbasis enthaltenen tatsächlichen Positives auch als solche erkannt wurden. Die Erkennungsrate der Negatives hat keinen Einfluss auf die Sensitivität. Eine hohe Sensitivität lässt sich somit „einfach“ erzielen, in dem man *alle* Instanzen immer als Positives klassifiziert.[25, S. 222]

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{2.22}$$

Die Spezifität nach Formel 2.23 bestimmt analog zur Sensitivität den Anteil der Negatives, die als solche klassifiziert wurden.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.23}$$

Ein Klassifikator, der alle Instanzen als Positives markiert, hat zwar eine Sensitivität von 100%, aber eine Spezifität von 0%. Ergeben zwei verschiedene Klassifikationsmodelle sehr ähnliche Genauigkeiten, hilft die Bestimmung der Sensitivität und der Spezifität bei der Auswahl des für den Anwendungsfall adäquateren Klassifikators. So ist beispielsweise bei der Bestimmung von schweren Krankheiten eventuell ein Klassifikator mit höherer Sensitivtät wünschenswert, um die Wahrscheinlichkeit zu minimieren, dass die entsprechende Krankheit nicht erkannt wird. [28] [25, S. 222]

3 Konzept zur Visualisierung von Schmerz Scores aus akustischen Signalen

In Kapitel 2.1 wurde vorgestellt, wie die Schmerzdiagnose bei Neugeborenen mit Hilfe von Pain Scales durch medizinisches Personal durchgeführt wird. Es wurde eine Reihe an Pain Scales vorgestellt und dabei insbesondere der Schmerzindikator „Weinen“ beleuchtet. Unabhängig von der konkret eingesetzten Pain Scale wird in jedem Fall das Baby für eine bestimme Zeitraum beobachtet, für jeden Indikator, wie das Weinen oder der Gesichtsausdruck, Punkte vergeben, diese Aufsummiert und aus der Summe der insgesamte Schmerz Score bestimmt.

In Kapitel 2.3 wurde vorgestellt, wie in der klassischen Schreiforschung mit Hilfe von Methoden der Signalverarbeitung das Weinen von Babys tiefergehend Analysiert wurde. Es wurde gezeigt, dass aus objektiv messbaren Eigenschaften des Weinens eines Babys Rückschlüsse auf dessen Zustand gemacht werden können.

Ziel dieser Arbeit ist der Entwurf eines Systems zur automatisierten Feststellung und Visualisierung von Pain Scores beliebiger Pain Scales mit dem Fokus auf den Schmerzindikator „Weinen“. Das System muss folgenden Anforderungen erfüllen:

1. Das System muss dazu in der Lage sein, aus den akustischen Eigenschaften des Weinens eines Babys den Schmerz Score bezüglich einer Pain Scale abzuleiten.
2. Das System muss dazu in der Lage sein, die abgeleiteten Schmerz Scores zu visualisieren.
3. Das System muss dazu in der Lage sein, beliebige Pain Scales einzubinden.
4. Die System muss dazu in der Lage sein, die Analyse auch bei nicht-optimalen akustischen Bedingungen durchzuführen.
5. Das System muss dazu in der Lage sein, die Analyse kontinuierlich durchzuführen.

Der **Input** des Systems ist folglich ein Audiosignal, welches kontinuierlich in das System gegeben wird. Der **Output** ist eine Visualisierung der abgeleiteten Pain Score, welche kontinuierlich erzeugt wird.

In Kapitel 3.1 wird zunächst ein Überblick über einige Veröffentlichungen gegeben, in denen Konzepte zur automatisierten Analyse und Auswertung kindlicher Lautäußerungen vorgestellt wurden. In Kapitel 3.2 wird daraufhin das in dieser Arbeit entworfene Konzept in Form einer Verarbeitungs-Pipeline vorgestellt. Diese Verarbeitungs-Pipeline kombiniert das Vorgehen bei der Schmerzdiagnostik auf Basis der Pain Scales, Methoden der klassischen Schreiforschung sowie einige Ideen der im folgenden Kapitel vorgestellten Veröffentlichungen.

3.1 Literaturüberblick

Der größere Teil der Veröffentlichungen, die sich in das Feld der Analyse von Audioaufnahmen Neugeborener einordnen lassen, stellten Algorithmen zur Klassifizierung einzelner Cry Units vor, entweder bezüglich der Weinursache (Hunger, Angst, Schmerz, ...) oder zur Diagnose bestimmter Krankheiten. Diese Methoden waren in den meisten Fällen nicht für die kontinuierliche Analyse geeignet, sondern hatten das Ziel, eine gegebenen Cry-Unit mit einer möglichst hohen Genauigkeit bezüglich des jeweiligen Sachverhaltes zu klassifizieren. Probleme wie Hintergrundrauschen, Berechnungsaufwand oder kontextuelle Informationen haben eine untergeordnete Rolle gespielt. Beispiele für solche Veröffentlichungen sind die von Abdulaziz et al. [53] oder Fuhr et al. [49].

Várallyay stellte in seiner Dissertation „Analysis of the Infant Cry with Objective Methods“ [51] Methoden zur automatisierten Analyse kindlicher Lautäußerungen vor. Das primäre Ziel der Dissertation war die Erforschung der Unterschiede zwischen den Lautäußerungen gesunder und tauber Neugeborener. Die Algorithmen zur automatisierten Analyse der Audiosignale waren ein „Nebenprodukt“ zur schnelleren Datenauswertung. Die Auswertung musste nicht kontinuierlich erfolgen. In der vorgestellten Verarbeitungs-Pipeline wurde das Eingangssignal in Zeitfenster weniger Millisekunden zerlegt und jedes Fenster nach Entscheidungsregeln als *stimmhaft* oder *nicht stimmhaft* markiert. Die stimmhaften Signalfenster wurden zu *Segmenten* zusammengefasst (welche in Kapitel 2.3.1 als Cry-Units bezeichnet werden). Auf Basis der Segmente wurden Auswertungen bezüglich des Zeitbereiches (Durchschnittliche Segmentlänge, Pausenlängen etc.), des Frequenzbereiches (Grund-Frequenz, Formanten-Frequenzen etc.) und des Melodieverlaufes angestellt. Analysiert wurden Audioaufnahmen von Babys mit einer Länge von 10 bis 100 s. Aus den Auswertungsergebnisse stellte Varallyay die wichtigsten Unterscheidungsmerkmale zwischen tauben und gesunden Babys fest. In der Dissertation [51] wird ein Überblick über das Vorgehen und die Ergebnisse gegeben. Die Verarbeitungsschritte wurden detaillierter in einzelnen Veröffentlichungen beschrieben, wobei der Autor dieser Arbeit nur den Zugriff auf einige dieser Veröffentlichungen erhalten konnte.

Cohen et al. haben 2012 in der Veröffentlichung „Infant Cry Analysis and Detection“ [6] ein System zur Analyse der akustischen Signale von Neugeborenen vorgestellt. Dieses System klassifizierte die Audiosignale in eine der drei Klassen *Cry, No Cry* und *No Activity*. Die Klasse *Cry* bezeichnet Lautäußerungen, die eine potentiell Gefahr für das Baby anzeigen, wie z.B. wie Schmerz oder Hunger. Die Klasse *No Cry* bedeutete, dass das Baby zwar Laute von sich gibt, diese aber keine potentielle Gefahr anzeigen. Die Klasse *No Activity* bezeichnete keinerlei Lautäußerung. Die Verarbeitungs-Pipeline wurde detailliert vorgestellt und war für die kontinuierliche Verarbeitung mit einer gewissen Verzögerungszeit spezialisiert. Das Signal wird in überlappende *Segmente* à 10 s zerlegt. Die Stimmaktivität in den Segmenten wird algorithmisch festgestellt. Wenn Aktivität vorliegt, wird das Segment in Sektionen à 1 s zerlegt und die Stimmaktivität für jede Sektion gemessen. Wird genügend Stimmaktivität in einer Sektion festgestellt, wird die Sektion in *Frames* à 32 ms zerlegt und Attribute für jeden Frame errechnet. Mit Hilfe von Entscheidungsregeln werden die Frames in *Cry, No-Cry* oder *No Activity* klassifiziert, wobei kontextuelle Informationen der umliegenden Frames mit einbezogen werden. Aus den Klassen der Frames wird auf die Klasse der Sektion geschlossen, und aus den Klassen der Sektionen auf die Klasse des Segmentes. Das System hat mit den Anforderungen dieser Arbeit gemeinsam, dass ebenfalls die kontinuierliche Verarbeitung im Vordergrund steht. Der Nachteil an dieser Methode ist,

dass die zeitliche längste Einheit, für die die Klassifizierung vorgenommen wird, unflexibel auf 10 s festgelegt ist. Daher müsste diese Verarbeitungs-Pipeline abgewandelt werden, um anstelle der Ableitung der drei genannten Klassen einen Pain Score ableiten zu können, die einen längeren Beobachtungszeitraum als 10 s benötigt.

Pal et al. haben 2006 in der Veröffentlichung „Emotion detection from infant facial experessions and cries“ [41] ein System vorgestellt, welches aus den akustischen Eigenschaften des Weinens die Emotion ableitet. Die zu erkennenden Emotionen sind *Traurigkeit, Wut, Hunger, Angst und Schmerz*. Es wird nicht erwähnt, ob die Analyse kontinuierlich oder nicht kontinuierlich erfolgt. Bei der Verarbeitung der akustischen Signale werden die Attribute *Grundtonhöhe* und die *Frequenz der ersten drei Formanten* extrahiert und mit einem Klassifizierungsalgorithmus klassifiziert. Es wurde nicht beschrieben, inwiefern die Eigenschaften aus kurzen Signalfenstern oder längeren Signalabschnitten errechnet werden, welche Vorverarbeitungsschritte angewandt werden und ob die Klassifizierung auf Ebene der Signalfenster oder über längere Zeitabschnitte hinweg geschieht.

Zamzi et al. haben 2016 in der Veröffentlichung „An Approach for Automated Multimodal Analysis of Infants' Pain“ [12] ein System zur automatisierten und kontinuierlichen multimodalen Analyse von Neugeborenen zur Ableitung des Schmerzes vorgestellt. Das System trägt den Namen *MPAS*. Der Schmerzgrad wird aus den Analyseergebnissen der monomodalen Schmerzindikatoren für *Gesichtsausdruck, Körperbewegung, Vitalfunktionen und Weinen* errechnet. Das System kommt der Aufgabenstellung dieser Masterarbeit am nächsten, da es ebenfalls um die Ableitung von Schmerz in einem multimodalen Verbund geht. Der Schmerz wurde hier „direkt“ abgeleitet, ohne den Weg über Pain Scales zu wählen. Während in der Veröffentlichung die Analyse der ersten drei genannten Schmerzindikatoren angekündigt wurde, wurden daraufhin die Methoden zur Analyse der akustischen Signale *nicht* erläutert. Auch die ersten Validierungsergebnisse beziehen sich nur auf den Gesichtsausdruck, die Körperbewegung und die Vitalfunktionen. Es ist nicht klar, ob die Miteinbeziehung akustischer Signale fallen gelassen wurde. Die Ausführungen konzentrieren sich dazu vermehrt auf die Methoden zur Kombination der Auswertungsergebnisse der monomodalen Schmerzindikatoren.

3.2 Verarbeitungs-Pipeline

In Kapitel 3.1 wurden verschiedene Konzepte vorgestellt, deren Fokus ebenfalls die Analyse und Auswertung von Audioaufnahmen kindlicher Lautäußerungen waren und somit der Aufgabenstellung dieser Arbeit ähneln. Keines der präsentierten Konzepte eignet sich, um mit nur leichten Anpassungen übernommen werden zu können: Entweder wurden die Verarbeitungsschritte nicht für die kontinuierliche Verarbeitung konzipiert [53] [49] [51], nicht genügen abstrahiert, um für andere Klassifizierungen als die ursprünglich geplanten abgewandelt werden zu können [6], oder die Verarbeitungs-Pipeline wurde nicht vorgestellt. [41] [12].

In dieser Arbeit wurde die folgende Verarbeitungs-Pipeline entworfen. Sie wird in in Abbildung 3.1 visualisiert.

1. **Input:** Ein Audiosignal, das möglicherweise kindliche Lautäußerungen enthält. Es wird kontinuierlich in das System gegeben.
2. **Vorverarbeitung** (engl. *Pre-Processing*) des Signals.

3. **Erkennung der Cry-Units**. Zunächst muss festgestellt werden, ob und wo in dem Signal kindliche Lautäußerungen vorhanden sind. Ein Algorithmus zur Feststellung von Stimmaktivität, bezeichnet als Voice Activity Detection, untersucht das Signal und markiert Cry-Units. Die gefunden Cry-Units bilden die Grundlage aller darauf folgenden Verarbeitungsschritte. Der vorgestellte Algorithmus kombinierte herkömmliche Methoden der Voice Activity Detection mit Ideen, die von Varallyay [51] vorgestellt wurden.
4. **Segmentierung** (engl *Segmenting*). Eine Pain Score wird nicht aus der Beobachtung einer einzigen, sondern einer Reihe von Cry-Units abgeleitet. In Kapitel 2.1.1 wurde gezeigt, dass bestimmte Pain Scores die Beobachtung über mehrere Minuten hinweg erfordern. Zu diesem Zweck werden die Cry-Units gruppiert. Da keines der in Kapitel 3.1 vorgestellten Konzepte Methoden zur Segmentierung einführte, wurde ein eigener Algorithmus für diese Aufgabe entworfen.
5. **Extrahierung von Eigenschaften und Ableitung der Schmerz Score** (engl. *Feature Extraction* und *Prediction of Pain Score*). Für jedes Segment werden Eigenschaften bezüglich des Weinens berechnet, wie zum Beispiel die durchschnittliche Tonhöhe, durchschnittliche Pausenlänge usw. Dabei werden Ideen der in Kapitel 2.3 vorgestellten klassischen Schreiforschung implementiert. Auf Basis dieser Eigenschaften wird die Pain Score abgeleitet.
6. **Output: Visualisierung** (engl. *Visualisation*) der abgeleiteten Schmerz Score. Es werden mehrere Varianten vorgeschlagen, bei der die Höhe des Schmerz Score in seinem zeitlichen Verlauf auf Ampelfarben abgebildet wird.

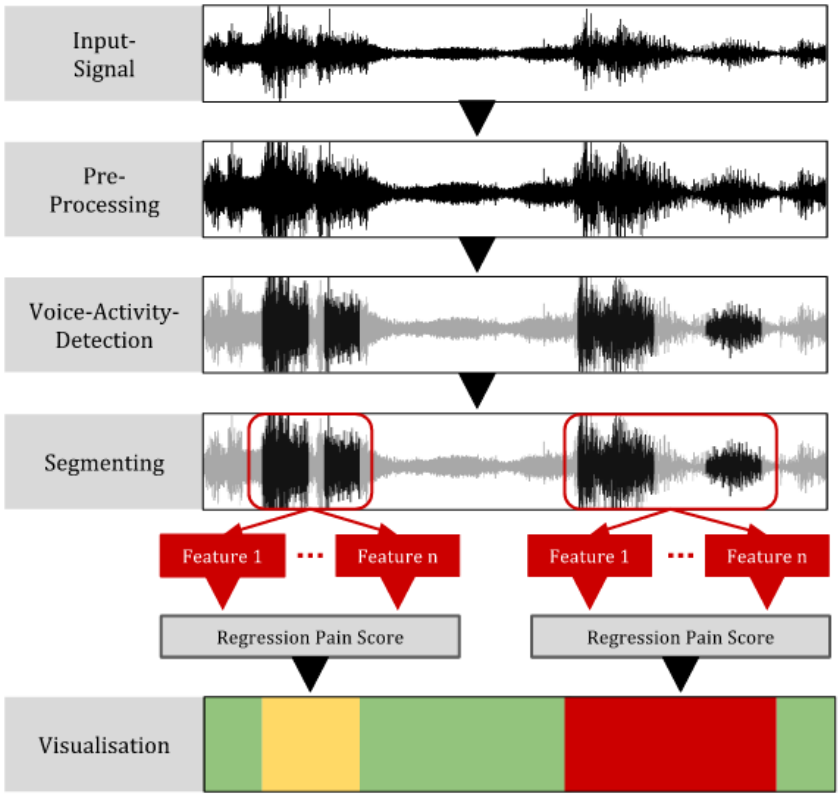

Abbildung 3.1: Überblick über die Verarbeitungs-Pipeline dieser Arbeit

4 Erkennung der Cry-Units

Der erste notwendige Schritt zur Ableitung einer Pain Score aus einem Audiosignal ist die Feststellung, ob in dem Signal überhaupt kindliche Lautäußerungen vorhanden sind. Das Ziel ist, in einem Audiosignal diejenigen Bereiche zu markieren, in denen Stimmaktivität vorhanden ist und daraufhin Beginn und Ende der einzelnen Cry-Units zu festzulegen. Abbildung 4.1 verdeutlicht diese Aufgabe an einem Beispiel: Der obere Graph zeigt in Schwarz ein Audiosignal. Die rote Linie, die das Signal überspannt, zeigt, welche Regionen Stimme enthalten, wobei 1$\hat{=}$ *stimmhaft* (engl. *voiced*) und 0$\hat{=}$ *nicht stimmhaft* (oder *Stille*, engl. *not voiced*). Das untere Schema zeigt, wie die stimmhaften Signalbereiche zu Cry-Units zusammengefasst wurden.

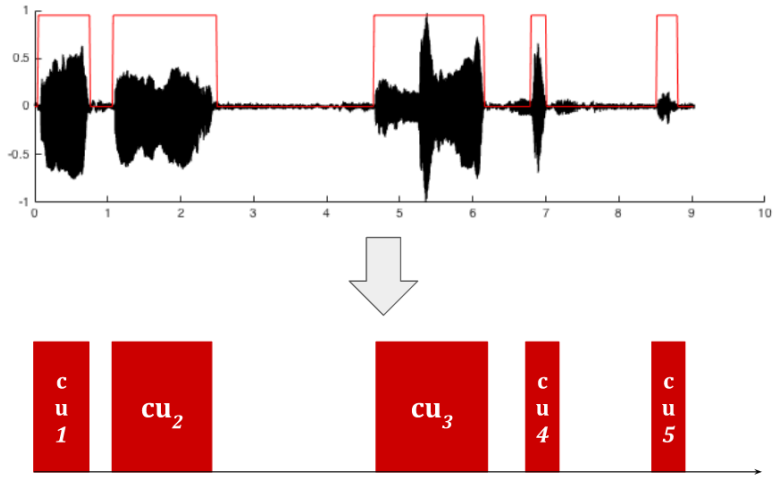

Abbildung 4.1: Markierung stimmhafter Bereiche in einem Audiosignal. Oben Schwarz: Das Eingangssignal $x[\,]$. Oben Rot: Klassifizierung in stimmhaft/Stille. Unten Rot: Die fünf erkannten Cry-Units.

In Kapitel 4.1 wird die Voice Activity Detection vorgestellt, das Feststellen des Vorhandenseins von Stimme in einem Signal. In Kapitel 4.2 wird besprochen, wie die stimmhaften Signalbereiche zu Cry-Units zusammengefasst werden. In Kapitel 4.3 wird ein Algorithmus vorgestellt, in dem Nachträglich inkorrekt erkannte Anfangs- und Endzeitpunkt von Cry-Units korrigiert werden.

4.1 Voice Activity Detection

Voice Activity Detection (kurz *VAD*) oder *Speech Detection* ist bei jeder Art der Sprachverarbeitung von Bedeutung: Im Mobilfunk wird sie beispielsweise eingesetzt, um die Zeitbereiche zu Erkennen, in denen die Teilnehmer sprechen und somit eine Übertragung stattfinden muss. Die größte Herausforderung von VAD-Systemen ist die robuste Erkennung

von Stimmaktivität auch bei starkem Hintergrundrauschen. Bis heute wurde keine „perfekte Lösung“ des Problems gefunden. [20, S. 1] [22, S. 1] [50, S. 1]

Der Grundlegende Aufbau eines VAD-Algorithmus ist wie folgt. Abbildung 4.2 visualisiert diesen Aufbau.

1. **Vorverarbeitung** (engl. Pre-Proccessing) des Signals.
2. **Windowing:** Unterteilung des Signals in (einander überlappende) Signalfenster.
3. **Extraktion von Eigenschaften** (engl. *Feature-Extraction*) aus jedem Signalfenster.
4. **Entscheidung** (engl. *Decision*) über die Präsens oder Abwesenheit von Stimme für jedes Signalfenster auf Grundlage der extrahierten Eigenschaften
5. **Decision-Smoothing**, das nachträgliche Hinzufügen oder Entfernen von Entscheidungen mit Hilfe kontextueller Informationen der umliegenden Entscheidungen.[20, S. 8 - 9] [22, S. 1 - 2]

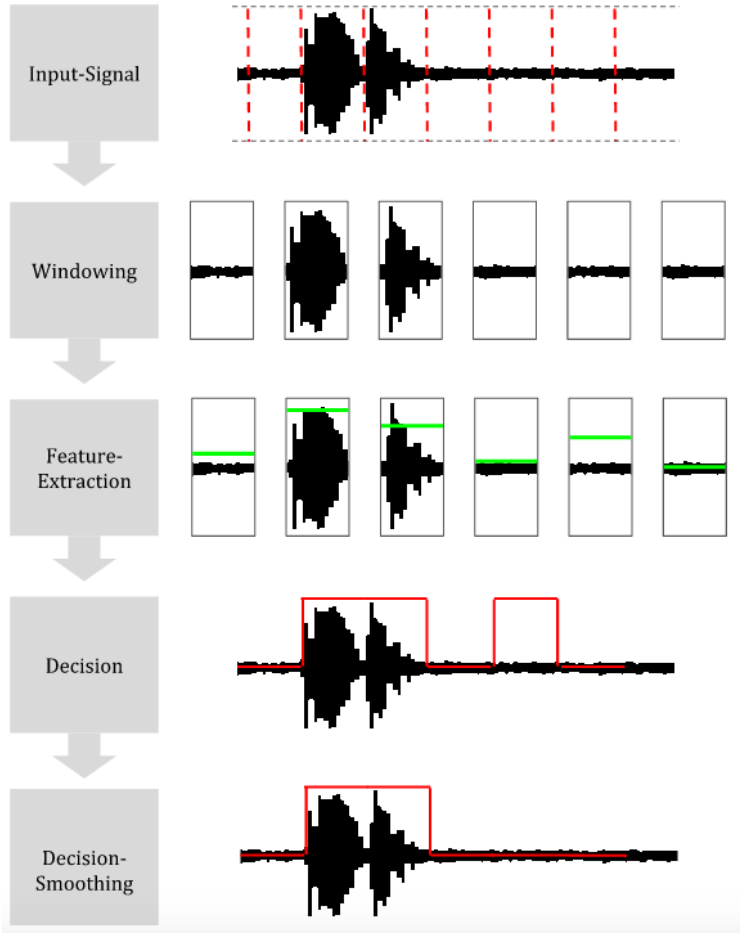

Abbildung 4.2: Aufbau eines VAD-Algorithmus

In Kapitel 4.1.1 werden die Methoden vorgestellt, die zur Voice Activity Detection erprobt wurden. In Kapitel 4.1.2 wird eine Simulationsstudie beschrieben, deren Ziel die Bestimmung derjenigen Methoden war, die sich für die VAD im speziellen Fall kindlicher Lautäußerungen am besten eignen. Kapitel 4.1.2 fasst die Ergebnisse zusammen.

4.1.1 Methoden

Die in diesem Kapitel vorgestellten Methoden kombinieren Ideen, die von Moattar et al. [30], Kristjansson et al. [50], Waheed et al. [24], Ahmadi et al. [46] und Shen et al.[21] vorgestellt wurden.

Vorverarbeitung

Bei der Vorverarbeitung wird das Signal so manipuliert, dass Störeinflüsse auf die darauf folgenden Verarbeitungsschritte minimiert werden. Welche Vorverarbeitung durchgeführt wird, ist Abhängig von der konkreten Aufgabenstellung. Dieser Schritt ist für die VAD optional. So setzten beispielsweise Ahmadi et al. [46] einen Bandpassfilter bei der Vorverarbeitung ein, während Moattar et al. [30] keine Vorverarbeitung anwendeten.

In dieser Arbeit sich für eine Vorverarbeitung entschieden, bei der das Signal hinsichtlich seiner Dynamik im Zeitbereich eingeschränkt wird. Dies ist ein typischer Vorverarbeitungsschritt bei Sprachaufnahmen. So wird vermieden, dass ein Signal eventuell zu leise ist, damit überhaupt Etwas darin erkannt werden kann. Einer der hauptsächlichen Gründe, warum ein Signal eine niedrige durchschnittliche Energie aufweisen kann, obwohl es maximal ausgesteuert wurde, sind sehr kurze Pegelspitzen, deren Pegel weit über dem Durchschnittspegel liegen und so eine weitere Erhöhung der Lautstärke verhindern. Da die Audiosignale, die in der in Kapitel 4.1.2 vorgestellten Simulationsstudie verwendet wurden, aus inhomogenen Quellen stammen und sehr unterschiedliche Lautstärken hatten, wurde so eine Angleichung der Signalenergien gewährleistet.

Die Dynamikeinschränkung wurde mit Hilfe eines Audio-Kompressors umgesetzt. Dieser verringert Signalspitzen, die über einen festgelegten *Schwellwert* (engl. *Threshold*) θ liegen, um ein festgelegtes *Verhältnis* (engl. *Ratio)* ρ. Ein Schwellwert von $\theta = 0.3$ mit einem Verhältnis von $\rho = 0.5$ bedeutet beispielsweise, dass alle Signalspitzen, die den Wert 0.3 über-, oder -0.3 unterschreiten, um 50% verringert werden. Der Wert eines Samples nach der Kompression $x_{comp}[n]$ ergibt sich somit nach Gleichung 4.1.

$$\text{comp}(x[n], \theta, \rho) = \begin{cases} \theta + (x[n] - \theta)\rho & \text{, wenn } x[n] > \theta \\ -\theta + (x[n] + \theta)\rho & \text{, wenn } x[n] < -\theta \\ x[n] & \text{sonst} \end{cases} \tag{4.1}$$

Die Amplituden hoher Signalspitzen werden so verringert, wodurch Headroom gewonnen wird, welcher anschließend bei der gleichmäßigen Erhöhung aller Amplituden zur Erhöhung der insgesamten Energie genutzt werden kann. Diese Erhöhung kann beispielsweise durch eine Normalisierung nach Gleichung 4.2 durchgeführt werden.

$$\text{normalize}(x[n]) = \frac{x[n]}{\max\{x[\,]\}} \tag{4.2}$$

Bei dem Kompressor, der in dieser Verarbeitungs-Pipeline zur Vorverarbeitung verwendet wird, werden Threshold und Ratio nach Formel 4.3 als Funktion des RMS-Wertes des

Signals berechnet. Der Parameter r_a gibt den Ziel-RMS-Wert an. Der RMS-Wert wird nach Formel 2.7 berechnet.

$$\theta(x[\,]) = \rho(x[\,]) = \left[\frac{\mathrm{RMS}(x[\,])}{r_a}\right]^2 \tag{4.3}$$

Die Vorverarbeitung wurde durchgeführt, indem 1.) die Kompression mit den Parametern nach Gleichung 4.3 und 2.) die Normalisierung nach Gleichung 4.3 durchgeführt wurde. Abbildung 4.3 zeigt ein Signal vor und nach der Vorverarbeitung nach diesem Prinzip. Um eine zu große Beeinflussung des Signals zu vermeiden, wurde ein Minimalwert für Threshold und Ratio von 0.4 festgelegt.

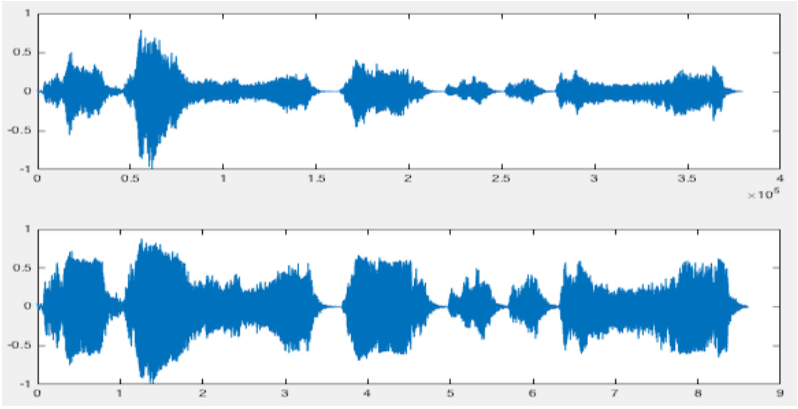

Abbildung 4.3: Ergebnis der Vorverarbeitung. Oben: Das Signal vor der Vorverarbeitung. Unten: Das Signal nach der Vorverarbeitung.

Diese Vorverarbeitung eignet sich nicht für ein kontinuierliches System, da Samples zur Berechnung des RMS-Wertes mit einbezogen werden, die zu einem Berechnungszeitpunkt in der Zukunft liegen. Das Hauptziel dieser Vorverarbeitung war die Herstellung ähnlicher Energieverhältnisse bei den Signalen, die in der Simulationsstudien in Kapitel 4.1.2 verwendet wurden. Damit diese Vorverarbeitung in einem kontinuierlichen System eingesetzt werden kann, wird die folgende Abwandlung vorgeschlagen:

- Komplettes Überspringen der Vorverarbeitung, in dem mit Hilfe eines manuell einstellbaren Lautstärkereglers eine ausreichende Signalenergie gewährleistet wird. Würde man beispielsweise anstelle der Analyse des Audiosignals das Gesicht des Neugeborenen mit einer Kamera untersuchen, würde man ebenfalls eine ausreichende Bildhelligkeit voraussetzen.
- Die Initialisierung des Kompressor mit "sanften Werten", wie zum Beispiel $\theta = \rho = 0.7$ und $\max\{x[\,] = 0.9\}$. Diese Parameter können nach der Beendigung eines Segmentes (Siehe Kapitel 5.1) auf Basis des RMS-Wertes des Segmentes aktualisiert und für die Vorverarbeitung der zukünftigen Werte eingesetzt werden.

Windowing

Angenommen, man führt die Voice Activity Detection für das Signal $x[\,]$ durch und kommt zu dem Schluss, dass in dem Signal (teilweise) Stimme enthalten ist. Ist das Signal mehrere Minuten oder sogar Stunden lang, ist allein aus der Entscheidung nicht ersichtlich, in

welchen Zeitbereichen genau Stimme enthalten ist, und in welchen nicht. Um die zeitliche „Auflösung“ zu erhöhen, wird das Signal in kürzere Zeitfenster zerlegt.

Nach der Vorverarbeitung wird diese Zerlegung mit Hilfe der in Kapitel 2.2.4 als *Windowing* bezeichneten Methode durchgeführt. Das Signal $x[\,]$ wird nach Gleichung 2.14 in die Signalfenster $x_0[\,], \ldots, x_m[\,]$ aufgeteilt. Die Zeitfenster werden zunächst im Zeitbereich belassen. Es wurde sich für die von Waheed et al. [24] vorgeschlagene Fensterlänge von 25 ms entschieden, als Kompromiss zwischen den von Moattar et al. [30] empfohlenen 10 ms und den von Ahmadi et al. [46] empfohlenen 40 ms. Die Fenster überlappen einander um 50%, das heißt 12.5 ms.

Die Entscheidung über das Vorhandensein von Stimme wird *einzeln* für jedes der Signalfenster $x_0[\,], \ldots, x_m[\,]$ durchgeführt. Grundlage für die Entscheidung ist eine Menge an Features, die für das jeweilige Signalfenster $x_i[\,]$ berechnet wird. Die Erforschung geeigneter Features ist eines der primären Forschungsgegenstände der VAD. In den folgenden Kapiteln 4.1.1 bis 4.1.1 wird eine Reihe an Features vorgestellt, die in dieser Arbeit zur VAD erprobt wurden. Jedes der in diesen Kapiteln besprochenen Features kann für ein Signalfenster $x_i[\,]$ à 25 ms berechnet werden und als Grundlage für eine Entscheidung dienen. Die Evaluation, welche Eigenschaften sich am besten zur Feststellung von Stimme im speziellen Fall von Neugeborenen eignen, folgt in Kapitel 4.1.2

Eigenschaften des Zeitbereiches

Im Zeitbereich wurden die beiden Eigenschaften*Root Mean Square* [*RMS*] und *Zero Crossing Rate* [*ZCR*] erprobt.

Moattar et al. [30] bezeichnen den Energiegehalt eines Signals als das für die VAD am häufigsten angewandte Attribut. Der RMS-Wert als Feature für ein Signalfenster wurde nach Gleichung 2.7 berechnet. Hintergrund ist, dass der Energiegehalt eines Stimmsignals typischerweise höher ist als der des Hintergrundrauschens. Bei geringem Signal/Rausch-Abständen ist diese Bedingung jedoch nicht immer gegeben. Als zweites Attribut des Zeitbereiches wurde die *Zero Crossing Rate* berechnet. Die ZCR nach Formel 4.4 gibt an, wie häufig ein Vorzeichenwechsel im Signal vorkommt. Eine höhere ZCR weist auf ein stimmloses Signal hin, da Rauschen typischerweise eine höhere ZCR als stimmhafte Signale aufweist. Problematisch ist dieses Kriterium bei Signalen, bei denen kein Hintergrundrauschen vorliegt, da sich dort eine ZCR von 0 ergibt.[46] Um den Wert in Relation zur Fensterlänge setzen zu können, wurde weiterhin die ZCR durch die Anzahl der Samples eines Signalfensters N geteilt.

$$\mathrm{ZCR}(x_i[\,]) = \sum_{0}^{N-1} |\mathrm{sng}(x_i[n]) - \mathrm{sng}(x_i[n-1])| \tag{4.4}$$

Eigenschaften der Autokorrelation

Neben den in Kapitel **??** genannten „einfachen“ Attributen des Zeitbereiches wurde die Autokorrelation zur VAD erprobt. Die Autokorrelation eignet sich, um Periodizität in einem Signal nachzuweisen. Wie in Kapitel 2.2.5 ausgeführt, weisen stimmhafte Signale eine tendenziell stärkeres periodisches Verhalten als das Hintergrundrauschen auf.

Bei der Autokorrelation wird ein Signal mit einer verzögerten Variante von sich selber korreliert. Gleichung 4.5 definiert die Autokorrelation des N-Sample langen Signalfensters $x_i[\,]$, verzögert um das Lag k.

$$\text{A-Corr}_k(x[\,]) = \sum_{n=k}^{N} x[n-k] \cdot x[n] \tag{4.5}$$

Da der Autokorrelationswert neben der Periodizität von der Signalenergie abhängig ist, ist eine Normalisierung des Wertes wünschenswert. Es gibt verschiedene Varianten dieser Normalisierung. Gleichung 4.6 definiert die „normalisierte Autokorrelation“, bei der der Autokorrelationswert durch die RMS-Werte des verzögerten und unverzögerten Signals normalisiert wird.[50]

$$\text{NA-Corr}_k(x[\,]) = \frac{\sum_{n=k}^{N} x[n-k] \cdot x[n]}{\sqrt{\sum_{n=1}^{N-k} x[n]^2} \cdot \sqrt{\sum_{n=k}^{N} x[n]^2}} \tag{4.6}$$

Das Autokorrelations-Signal $a[\,]$ wird erstellt, indem die normalisierte Autokorrelation für verschiedene $k = k_{min}, \ldots, k_{max}$ angewandt wird, wie Gleichung 4.7 definiert.

$$a[\,] := \bigvee_{k=k_{min}}^{k_{max}} : \; a[k] = \text{NA-Corr}_k(x[\,]) \tag{4.7}$$

Ein hoher Wert des Signals $a[\,]$ an der Position k spricht für eine ausgeprägte Periodizität des Signals mit der Frequenz $f = f_s/k$. Es ist üblich, den Bereich $[k_{min}, k_{max}]$ so einzuschränken, dass die Autokorrelation nur für den Frequenz-Raum durchgeführt wird, in dem man Periodizität erwartet.[50]

In Bezug auf die VAD wurde die Autokorrelation wird als Methode genutzt, um die beiden Attribute *höchste Autokorrelationsspitze* [*aMax*] und *Anzahl der Autokorrelationsspitzen* [*aCount*] zu berechnen. Beide Eigenschaften wurden von Kristjansson et al. [50, S. 1 - 2] zur VAD beschrieben. Die *höchste Autokorrelationsspitze* wird in Formel 4.8 definiert und bestimmt die höchste Magnitude im Autokorrelationssignal. Ein stimmhaftes Signal hat aufgrund seiner Periodizität erwartungsgemäß einen höheren [*aMax*]-Wert als Rauschen.

$$\text{aMax}(x_i[\,]) = \max_k \text{mag}\{\text{NA-Corr}_k(x_i[\,])\} \tag{4.8}$$

Die *Anzahl der Autokorrelationsspitzen* wird nach Formel 4.9 berechnet. Das Feature gibt an, wie viele Signalspitzen im Autokorrelationssignal enthalten sind. Rauschen erzeugt höhere [*aCount*]-Wert als stimmhafte Signale, bedingt durch die vielen zufällig entstehenden Periodizitäten.

$$\text{aCount}(x_i[\,]) = \operatorname*{count}_k \text{mag}\{\text{NA-Corr}_k(x_i[\,])\} \tag{4.9}$$

Aus Kapitel 2.3.1 ging hervor, dass die Grundfrequenz der Stimme von Neugeborenen zwischen 250 und 2000 Hz liegt, weshalb auch nur in Lags dieses Bereichs bei der Berechnung beider Features verwendet wurden.

Eigenschaften des Frequenzbereiches

Aus dem Frequenzbereich wurden die drei Eigenschaften *unnormalisierte spektrale Entropie* [$SEnt_u$], *normalisierte spektrale Entropie* [$SEnt_n$] und *dominanteste Frequenzkomponenten* [f_{dom}] erprobt.

Als Vorbereitungsschritt muss das Signalfenster des Zeitbereiches $x_i[\,]$ in den Frequenzbereich $X_i[\,]$ transformiert werden. Die Berechnungsvorschrift ist $X_i[\,] = \mathrm{DFT}\{(w[\,] \cdot x_i[\,])\}$. Wird diese Transformation für alle Signalfenster $x_0[\,], \ldots, x_m[\,]$ eines Signals durchgeführt, entspricht dies der in Kapitel 2.2.4 vorgestellten Short Time Fourier Transformation. Es wurde eine 2048 Punkte Lange FFT und eine Hamming-Window als Fensterfunktion $w[\,]$ verwendet.

Kristjansson et al. [50, S. 2] haben die *spektrale Entropie* zur Voice Activity Detection beschrieben. Dabei wird das Spektrum des Frequenzfensters $X_i[\,]$ als Wahrscheinlichkeitsverteilung betrachtet. Die Entropie als Maß zur „Unreinheit“ wurde in Kapitel 2.4.1 erläutert. Die *normalisierte spektrale Entropie* wird nach der Formel 4.11 berechnet. Das Signal $px_i[\,]$ ergibt sich durch die Normalisierung des N-Punkte langen Spektrums nach Formel. Bei der normalisierten spektralen Entropie ist zu erwarten, dass Frequenzfenster ohne Stimme einen höheren Wert aufweisen als Fenster mit Stimme.4.10.

$$px_i[n] = \frac{X_i[n]}{\sum_{k=1}^{N} X_i[k]} \tag{4.10}$$

$$\mathrm{SEnt}_n(px_i[\,]) = -\sum_{k=1}^{N} px_i[k] \cdot \log(px_i[k]) \tag{4.11}$$

Neben der von Kristjansson et al. [50] vorgestellten normalisierten spektralen Entropie wurde zusätzlich die *unnormalisierte Spektrale Entropie* nach Formel 4.12 berechnet. Bei dieser wird das Spektrum nicht normalisiert, das heißt, es gilt $px_i[k] = X_i[k]$. Somit hat die Energie des Signals einen größeren Einfluss den Wert des Attributes. Dabei ist zu erwarten, dass Signalfenster mit Stimme einen höheren Wert aufweisen als Rauschen.[1]

$$\mathrm{SEnt}_u(X_i[\,]) = -\sum_{k=1}^{N} X_i[k] \cdot \log(X_i[k]) \tag{4.12}$$

In die Berechnungen wurden nur die Frequenzen im Bereich von 250 - 8000 Hz mit einbezogen, da nach Kapitel 2.3.1 die tiefst mögliche Frequenz der Stimme eines Babys bei 200 Hz liegt und nach Shen et al. [21] die Stimme keine Informationen oberhalb von 8000 Hz enthält.

Moattar et al [30, S. 2550] haben die *dominanteste Frequenzkomponente* zur Voice-Activity-Detection vorgestellt. Für jedes Frequenzfenster $X_i[\,]$ wird diejenige Frequenz nach Formel 4.13 berechnet, welche die höchste Amplitude hat. Es wird dabei, im Gegensatz zur spektralen Entropie, der gesamte Frequenzraum betrachtet. Ein stimmhaftes Signal

[1] Kristjansson et al [50, S. 2] verwenden zur Entropie-Berechnung den Logarithmus zur Basis 10, anstatt zur Basis 2. Es ist nicht klar, ob es sich dabei um einen Fehler handelt. In dieser Arbeit wurde, wie in dem Paper beschrieben, ebenfalls der Logarithmus zur Basis 10 verwendet!

hat typischerweise eine höhere f_{dom} als ein stimmloses Signal, bedingt durch die hohe Amplitude der Grundfrequenz.

$$f_{dom}(X_i[\,]) = \arg\max\{X_i[\,]\} \tag{4.13}$$

Eigenschaften des Cepstrums

Das Cepstrum wird nach Gleichung 4.14 als die inverse DFT des Logarithmus des Magnitudensignals des Frequenz-Bereiches definiert.[14, Cepstral analysis]

$$c[\,] = \text{iDFT}\Big\{ \log\Big(\,|\, \text{DFT}\{x[\,]\}|\, \Big)\Big\} \tag{4.14}$$

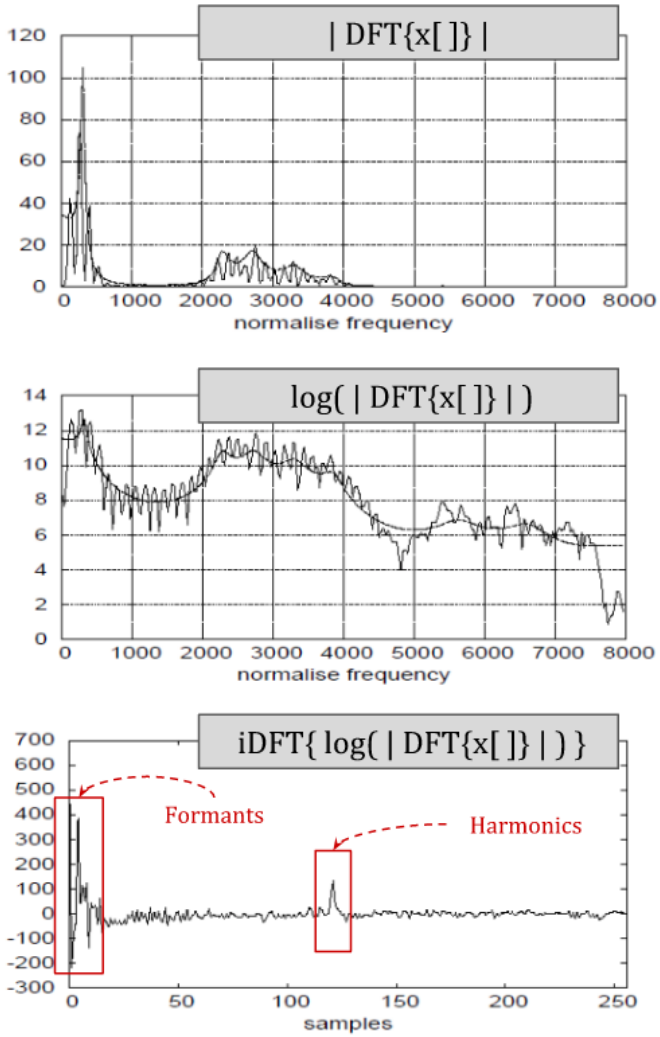

Abbildung 4.4: Berechnung des Cepstrums. (nach: [14, Cepstral Analysis, S. 3])

Das Vorgehen wird mit Hilfe des Beispiels aus Abbildung 4.4 erläutert. $|\,\text{DFT}\{x[\,]\}|$ zeigt das Spektrum eines „typischen stimmhaften“ Signals $x[\,]$. Es sind die in Kapitel 2.2.5 erläuterten für ein stimmhaftes Signal typischen harmonischen Obertöne zu sehen,

welche mit steigender Frequenz an Amplitude verlieren. Durch das logarithmieren des Spektrums $\log\left(\,\left|\,\mathrm{DFT}\{x[\,]\}\right|\,\right)$ wird die Dynamic des Frequenzbereiches verringert und somit der Amplitudenverlust der höheren Obertöne verringert. Nun stellt man sich vor, dieses Spektrum wäre ein Signal des Zeitbereiches. Dieses Signal würde man als ein annährend periodisches Signal mit einer Amplituden-Modulation interpretieren, das heißt ein Signal mit hoher Frequenz, addiert mit einem Signal mit nierdiger Frequenz. Um diese beiden Komponenten voneinander zu trennen, müsste man eine weitere DFT anwenden. Diese DFT kommt in dem Fall einer inversen DFT gleich, da das Phasen-Signal verworfen wurde. Man erwartet in diesem „Spektrum vom Spektrum“ einen Peak im „oberen Frequenzbereich“, bedingt durch die harmonischen Oberwellen, sowie einen Peak im „unteren Frequenzbereich“, bedingt durch die Formanten.[14, Cepstral analysis]

Der Bereich dieser „Fouriertransformation der Fouriertransformation“ wird als *Cepstrum* bezeichnet. Cepstrum ist ein ein Wortspiel, welches durch die Umkehrung der ersten vier Buchstaben des Wortes SSpectrumëntsteht. Die Unabhängige Variable des Cepstrum wird als *Quefrency* bezeichnet. Damit wird verdeutlicht, dass die unabhängige Variable des Cepstrum zwar mathematisch betrachtet die Zeit darstellt, jedoch als Frequenz interpretiert wird.[14, S. 7]

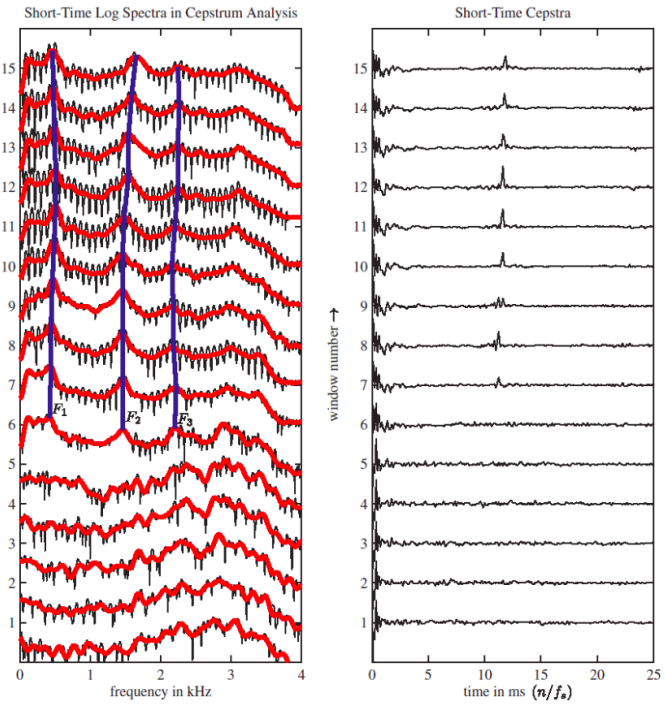

Abbildung 4.5: Aufkommen eines Peaks im oberen Quefrency-Bereich bei stimmhaften Signalfenstern. [14, Cepstral Analysis, S. 17]

Ein Auftauchen eines Peaks im oberen Quefrency-Bereich $> 3\,\mathrm{ms}$ spricht für das Vorhandensein von harmonischen Obertönen im Signal, wie sie durch Stimme erzeugt werden. Abbildung 4.5 verdeutlicht das Prinzip an einem Beispiel. Zu sehen ist die STFT eines Signals mit einer Fensterlänge von 50 ms und einer Hopsize von 12.5 ms. Links wird das logarithmierte Spektrum abgebildet, rechts das Cepstrum. Die Frames 1 bis 5 sind stimmlos, die Frames 8 bis 15 sind stimmhaft, und die zwischen-Frames eine Mischung. Man sieht das Aufkommen eines Peaks bei einer Quefrency $q = 12\,\mathrm{ms}$.[14, S. 16]

Abbildung 4.6 verdeutlicht, wie eine Grundfrequenz f_0 im Zeitbereich einen Peak im Cepstrum erzeugt. So weist ein Peak an bei der Quefrency q auf eine Grundfrequenz von $f_0 = f_s/q$ hin.[44]

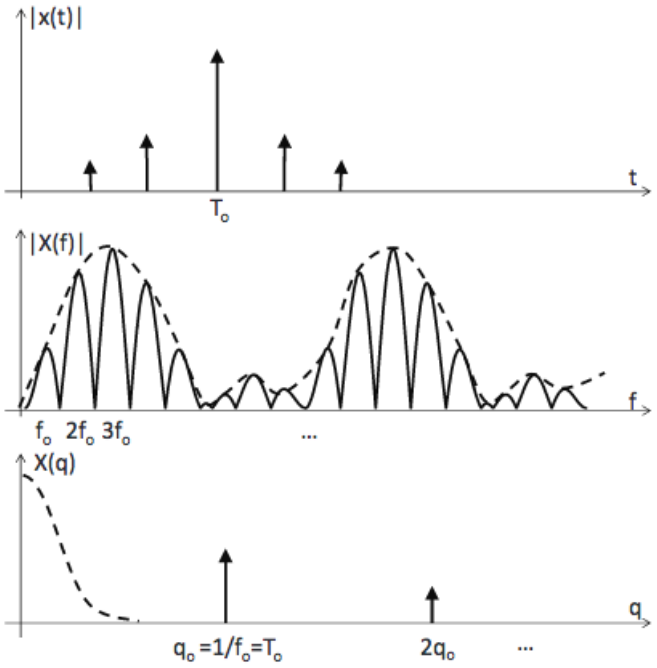

Abbildung 4.6: Feststellung der Grundfrequenz aus dem Cepstrum. [44]

In Bezug auf die Voice Activity Detection wurde das Cepstrum genutzt, um die beiden Features *Obere Cepstrum-Spitze* [$Ceps_{mag}$] und *Quefrency der oberen Cepstrum-Spitze* [$Ceps_{loc}$] zu berechnen.

Ahmadi et al. [46] sowie Kristjansson et al.[50] schlagen vor, die Höhe der Magnitude eines Peaks im oberen Bereich des Cepstrums als Maß für die Stimmhaftigkeit des Signals einzusetzen. Formel 4.15 definiert die Berechnung. $c_i[\,]$ ist das Cepstrum des i-ten Frequenzfensters $X_i[\,]$. Wie in Kapitel 2.3.1 erläutert, liegt die Grundfrequenz bei kindlichen Lautäußerungen zwischen 200 und 2000 Hz, was einem Quefrency-Bereich von 5 - 40 ms entspricht. Folglich werden bei der Berechnung nach Formel 4.15 nur Quefrency-Werte in diesem Bereich betrachtet.

$$Ceps_{mag}(c_i[\,]) = \underset{q=q_{min}\ldots,q_{max}}{\text{maxmag}} \{\, c[q] \,\} \tag{4.15}$$

Als zweites Attribut, welches auf dem Cesptrum basiert, wurde die Quefrency der höchsten Amplitude des oberen Cepstrum-Bereiches nach Formel 4.16 berechnet. Bei Signalfenstern ohne Stimme ist es wahrscheinlicher, dass sich die höchste Amplitude am Mindest- oder Maximalwert des durchsuchten Quefrency-Bereiches befindet.

$$Ceps_{loc}(c_i[\,]) = \arg\max\{\, c[\,] \,\} \tag{4.16}$$

Differenz-Feature

Abbildung 4.7 visualisiert alle Attribute, die in den Kapiteln 4.1.1 bis 4.1.1 vorgestellt wurden. Der oberste Plot zeigt das Audiosignal aus Abbildung 4.1 mit einem Signal/Rausch-Abstand von 20 dB. Der rote Graph über dem Plot klassifiziert die Zeitbereiche in $1 \mathrel{\hat{=}}$ *stimmhaft* und $0 \mathrel{\hat{=}}$ *nicht stimmhaft*. Alle darunter liegenden Plots zeigen den zeitlichen Verlauf der entsprechenden Attribute.

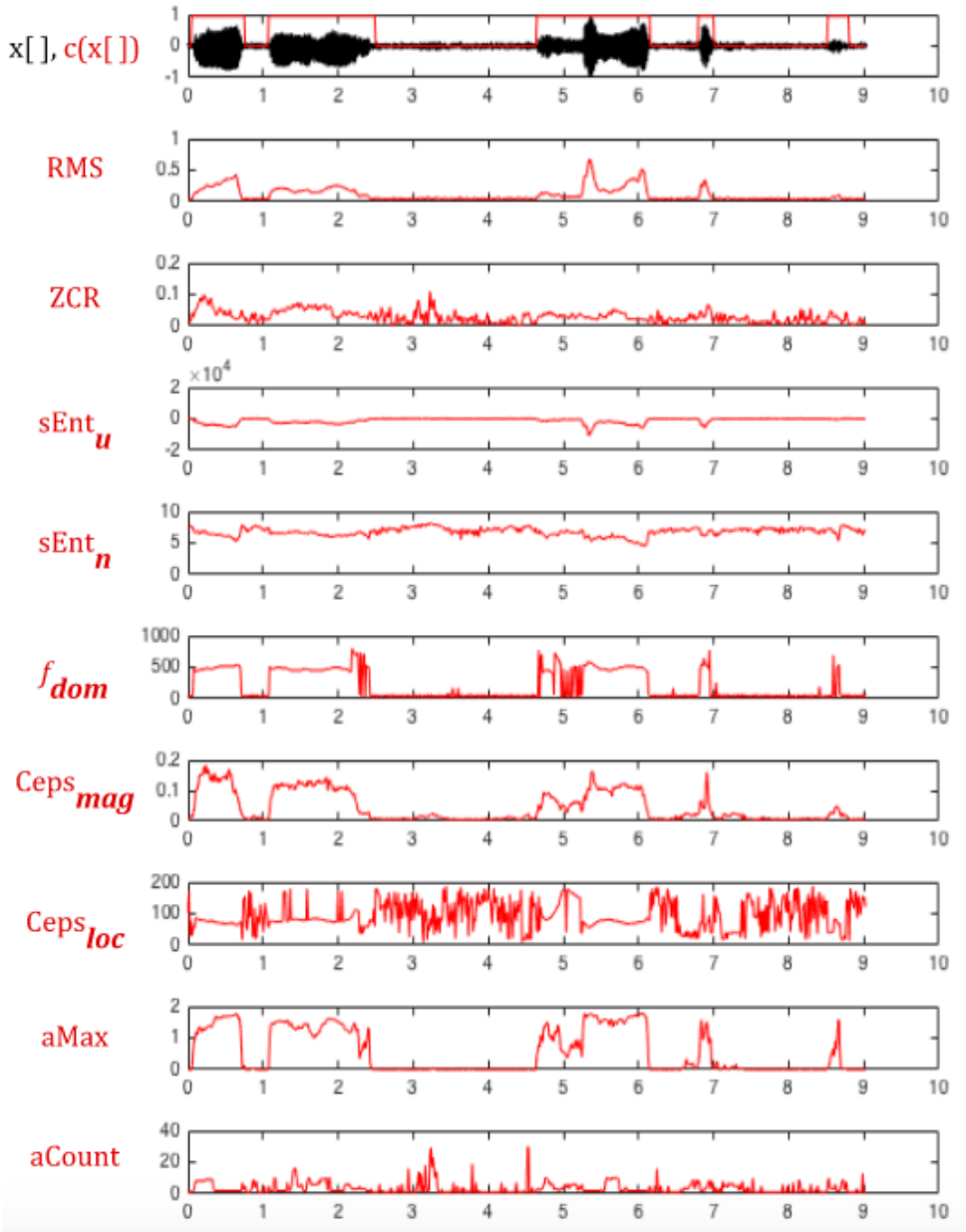

Abbildung 4.7: Übersicht über alle Features, die für die Voice Activity Detection erprobt wurden.

Abbildung 4.8 zeigt den zeitlichen Verlauf des RMS-Features im Detail. (A) zeigt das verhalten des *RMS*-Attributes bei einem Signal/Rauschabstand von 50 dB. Die stimmlosen Zeiträume haben einen weitaus niedrigeren RMS-Wert als die Zeiträume mit Stimme. In

(B) ist das selbe Signal mit einem Signal/Rauschabstand von 3 dB zu sehen. Nun liegen die RMS-Werte der stimmlosen Bereiche nur noch knapp unter denen des Sprachsignals. Zu sehen ist, dass starkes Hintergrundrauschen ähnlich hohe Feature-Werte erzeugen kann wie die Stimme.

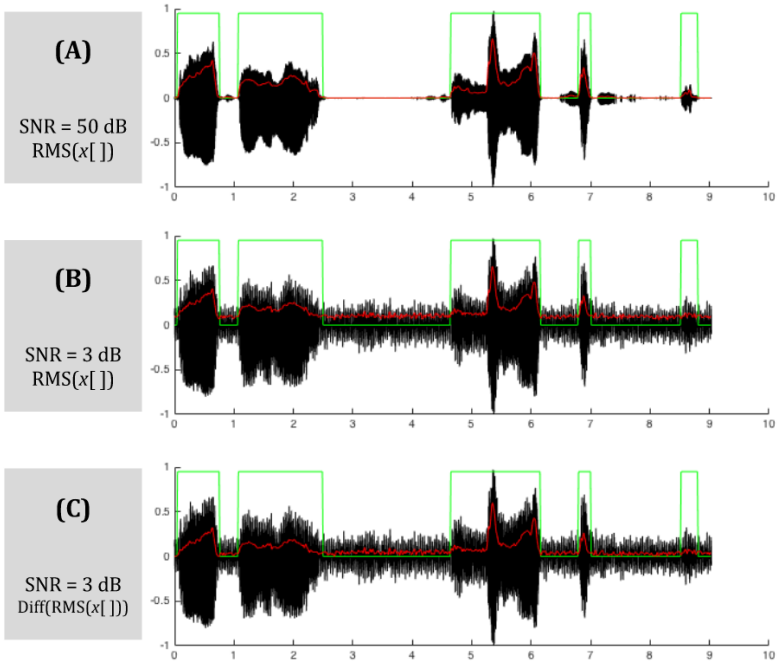

Abbildung 4.8: Das RMS-Feature bei verschiedenen Signal/Rausch-Abständen. Schwarz: Eingangs-Signal $x[\,]$. Grün: Klassifizierung in Stimmhaft/Stille. Rot: Feature-Wert.

Moattar et al [30] und Waheed et al [24] präsentierten die Idee, den Wert des jeweiligen Attributes zu messen, der in den stimmlosen Bereichen durch das Hintergrundrauschen erzeugt wird. Es kann davon ausgegangen werden, dass die ersten Signalfenster eines Signals stimmlos sind, und der Feature-Wert des Rauschens somit anhand dieser Fenster bestimmt werden kann. Bei einer langanhaltenden und kontinuierlichen Analyse können sich sowohl der Signal/Rausch-Abstand als auch die Qualität des Rauschens ständig ändern, weshalb die von den stimmlosen Bereichen erzeugten Attributwerte regelmäßig aktualisiert werden müssen. Es kann weiterhin davon ausgegangen werden, dass die Länge einer Cry-Unit eine bestimmte Länge t_{max} nicht überschreiten kann, bevor das Baby Luft holen muss und somit zumindest ein stimmloses Zeitfenster entsteht, welches das Hintergrundrauschen enthält. Zeskind et al. [40, S. 325] haben $t_{max} = 4.75\,\mathrm{s}$ bestimmt. In einem Zeitbereich $t > t_{max}$ muss somit zumindest ein Feature-Wert enthalten sein, der durch stimmlose Signalteile erzeugt wird.

Auf Basis dieser Überlegung wurde das *Differenz-Feature* $\mathrm{Diff_t}(\mathrm{Feat}(x_i[\,]))$ nach Formel 4.17 definiert als die Differenz zwischen einem aktuell gemessenen Attributwerte und dem geringsten Attributwerte, welcher im vergangenen Zeitbereich t gemessen wurde. $\mathrm{Feat}(x_i[\,])$ bezeichnet dabei einen beliebigen Feature-Wert des Signalfensters $x_i[\,]$, t_{xi} die Länge eines Signalfensters in Sekunden (in diesem Fall 25 ms), und t der in der Vergangenheit

zu durchsuchende Zeitbereich in Sekunden $> t_{max}$. In Abbildung 4.8 wird in (C) das Differenz-Feature für den RMS-Wertes gezeigt.

$$\text{Diff}_t(\text{Feat}(x_i[\,])) = \text{Feat}(x_i[\,]) \; - \min_{k=i-z\ldots i}\{\text{Feat}(x_k[\,])\}, \qquad z = \frac{2 \cdot t}{t_{xi}} \tag{4.17}$$

In dieser Arbeit wurde $t = 5\,\text{s}$ festgelegt. Es ist zu beachten, dass die Attribute *ZCR, SEnt*$_u$ und *aCount* zur Berechnung des Differenz-Features bezüglich ihres Vorzeichens invertiert werden müssen, da bei Ihnen ein niedriger an Stelle eines hohen Wertes stimmhafte Signale anzeigen. Das einzige Attribut, für dass die Berechnung des Differenz-Features keinen Sinn macht, ist das *Ceps*$_{loc}$-Attribut, da es bei stimmlosen Signalabschnitten sowohl einen höheren als auch einen niedrigeren Wert annehmen kann.

Entscheidung

Die einfachste Variante, um zu Entscheiden, ob ein Signalfenster $x_i[\,]$ stimmhaft ist, ist, eines der vorgestellten Features als Entscheidungsgrundlage zu wählen und einen festen Grenzwert θ festzulegen, bei dessen Über- oder Unterschreitung das Fenster als stimmhaft klassifiziert wird. Die Entscheidung lässt sich als Klassifizierungsfunktion C definieren, welche ein Signalfenster abbildet auf $Y = \{1 \mathrel{\hat{=}} \text{stimmhaft}, 0 \mathrel{\hat{=}} \text{nicht stimmhaft}\}$, das heißt $C(x[\,]) \mapsto \{1, 0\}$. Die Klassifizierungsfunktion nimmt dann die folgende Form an:

$$C(x_i[\,]) = \begin{cases} 1 & \text{, wenn } \text{Feat}(x_i[\,]) > \theta \\ 0 & \text{, sonst} \end{cases} \tag{4.18}$$

Abbildung 4.9 verdeutlicht das Prinzip an einem Beispiel. Das Feature, dass als Entscheidungsgrundlage genutzt wird, ist der *RMS*-Wert. Ein Grenzwert von $\theta = 0.18$ würde in diesem Fall eine weitestgehend richtige Klassifizierung gewährleisten.

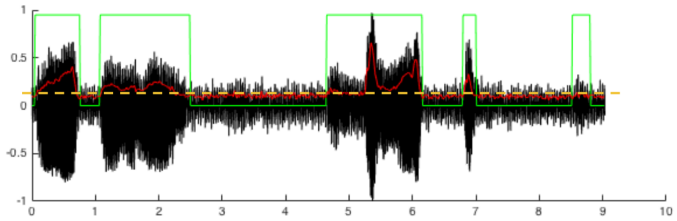

Abbildung 4.9: Thresholding eines Features. Schwarz: Das Eingangssignal $x[\,]$. Grün: Klassifizierung in Stimmhaft/Stille. Rot: RMS-Feature. Orange: Grenzwert

Die Frage ist, wie man diesen Grenzwert findet. In einigen Veröffentlichungen, so wie der von Moattar et al. [30] wird empfohlen, den Grenzwert experimentell festzustellen. In dieser Arbeit wurde sich für den Ansatz einer automatisierten, datengetriebenen Suche der Grenzwerte mit Hilfe des in Kapitel 2.4.1 vorgestellten *C4.5*-Algorithmus entschieden. Der Grundgedanke ist wie folgt: Angenommen, man hat einen Trainingsdatensatz zur Verfügung, bestehend aus einer Menge an Signalen mit Aufnahmen von kindlichen Lautäußerungen, in denen manuell die stimmhaften Signalabschnitte markiert wurden. Diese Signale zerlegt man nach den besprochenen Methoden in kürzere Signalfenster und berechnet für jedes Signalfenster das Feature, dessen Grenzwert man sucht. Jedem Signalfenster versieht man außerdem mit dem Klassenlabel $y \in Y$ auf Basis der manuellen Markierungen. Ein

Signalfenster $x_i[\,]$ wird so zu einem *Example* $e_i = (\text{Feat}(x_i[\,]), y_i)$. Nun kann der *C4.5*-Algorithmus verwendet werden, um für das Feature den Grenzwert zu finden, der für diese Datenbasis die Klassifizierung mit der höchsten Genauigkeit vornimmt. Voraussetzung dafür ist, dass der *C4.5*-Algorithmus gezwungen wird, einen Entscheidungsbaum mit einer maximalen Tiefe von 1 zu erzeugen, damit auch nur ein Grenzwert gesucht wird. Ist das Feature, für das der Grenzwert gesucht wird, ein Differenz-Feauture (siehe Kapitel 4.1.1), so lässt sich dieser Grenzwert als ein adaptiver Grenzwert interpretieren, der sich an den aktuellen SNR anpasst.

Die Verwendung des *C4.5*-Algorithmus bringt die Möglichkeit mit sich, eine beliebige Menge an Features in die Klassifizierung mit einzubeziehen. Die Klassifizierung muss also nicht zwingend auf Grundlage nur eines Attributes geschehen. Angenommen, es werden die beiden Features Ceps_{mag} und der RMS-Wert für jedes Signalfenster berechnet. Der *C4.5*-Algorithmus würde nun für einen Trainingsdatensatz einen Entscheidungsbaum konstruieren, bei dem durch das hierarchische Setzen von Grenzwerten die Klassifikation vorgenommen wird. Listing 4.1 zeigt ein Beispiel für eine so resultierende, fiktive Klassifizierungsfunktion.

Listing 4.1: Beispiel eines CART-Entscheidungsbaums

```
if Ceps_mag(x_i[]) > 0.2
|    if RMS(x_i[]) < 0.13
|    |    C(x_i[]) = 0
|    |else
|    |    C(x_i[]) = 1
|else
|     C(x_i[]) = 1
```

Wird dem *C4.5*-Algorithmus eine größere Menge an Attributen zur Verfügung gestellt, auf deren Basis der Klassifikator entworfen wird, wird der Entscheidungsbaum implizit zeigen, welche Features für die VAD besser geeignet sind, da sie höher im Entscheidungsbaum stehen werden. Es besteht jedoch die Gefahr, dass der Klassifikator in ein lokales Maximum gelaufen ist und eine suboptimale Auswahl an Features durch den Algorithmus getroffen wurde. Außerdem besteht die Gefahr von Overfitting, insofern Entscheidungsbäume mit beliebiger Tiefe zugelassen werden.

4.1.2 Simulations-Studie

In Kapitel 4.1.1 wurde eine Übersicht über die Methoden gegeben, die in dieser Arbeit zur Voice Activity Detection erprobt und/oder eingesetzt wurden. Neben Methoden zur Vorverarbeitung und zum Windowing wurde eine Reihe an Features vorgestellt, die für ein Signalfenster als Entscheidungsgrundlage für die Erkennung von Stimme dienen können. Schlussendlich wurde argumentiert, wie der *C4.5*-Algorithmus genutzt werden kann, um einen Klassifikator zu entwerfen, der die Entscheidung über das Vorhandensein von Stimme in einem Signalfenster auf Grundlage einer Menge von Features fällt. Dafür wird ein Datensatz zum Training des *C4.5*-Algorithmus benötigt.

Das Ziel ist es nun, auf Grundlage dieser Methode den tatsächlichen Klassifikator zu finden. Der Klassifikator soll dabei die folgenden Bedingungen erfüllen:

- Der Klassifikator soll eine möglichst kleine Anzahl an Features verwenden, da es zu aufwendig ist, alle vorgestellten Features nur für die VAD in einem kontinuierlichen System zu berechnen.

- Der Klassifikator soll eine möglichst hohe Genauigkeit erzielen, unabhängig von Stärke und Qualität des Hintergrundrauschens.

Das Vorgehen zum finden des Klassifikators war wie folgt:

1. Es wurde ein Menge an Trainingsdatensätzen erstellt, indem ...
   1.1. ... in einer Menge an Audioaufnahmen weinender Babys manuell die stimmhaften Signalabschnitte markiert wurden,
   1.2. ... die Signale vorverarbeitet, in Signalfenster zerlegt und für jedes Signalfenster alle vorgestellten Features berechnet wurden,
   1.3. ... woraus Datensätze erzeugt wurden, wobei jeder Datensatz nur eine bestimmte Untermenge der Features zur Verfügung gestellt bekam.
2. Auf Basis jedes Datensatzes wurde mit Hilfe des *C4.5*-Algorithmus ein Klassifikator erzeugt.
3. Jeder Klassifikator wurde bezüglich seiner Genauigkeit gegen eine Menge an Trainingsdatensätze unterschiedlicher Signal/Rausch-Abstände evaluiert.
4. Schlussendlich wurde sich für einen Klassifikator auf Grundlage der Evaluationsergebnisse entschieden.

In Kapitel 4.1.2 wird die Erstellung des Datensatzes erläutert. In Kapitel 4.1.2 wird das Vorgehen beim Training des *C4.5*-Algorithmus ausgeführt, und in Kapitel 4.1.2 die Ergebnisse ausgewertet.

Erstellung der Datensätze

Der zeitliche Rahmen dieser Arbeit ermöglichte es nicht, selber Audioaufnahme von Babys zu machen. Daher wurden sechs Audioaufnahmen mit dem Weinen verschiedener Babies unterschiedlicher Qualität und Intensität von der freien Online-Sound-Bibliothek `https://www.freesound.org/` heruntergeladen und zu Segmenten à 10 s beschnitten. Es handelt sich um weitestgehend rauschfreie Aufnahmen, die von verschiedenen Babys stammen. In den Audiosignalen wurden manuell die Zeitbereiche markiert, welche Stimme enthalten.[2]

Weiterhin wurden drei verschiedene Rauschsignale heruntergeladen. Es handelt sich um „realistische“ Atmosphären von Krankenhäusern. Jedes der sechs Audioaufnahmen der Babys wurde mit jedem der drei Rauschsignale überlagert, einmal mit einem Signal/Rausch-Abstand von 50 dB („fast unhörbares Rauschen“), und einmal mit einem Signal/Rausch-Abstand von 3 dB („starkes Rauschen“). Außerdem wurde ein siebte Aufnahme eines Babys heruntergeladen, welches mit einem vierten Rauschsignal mit einem SNR von 7 dB überlagert wurde. Dieses Signal spielte eine Sonderrolle, da es nur zur Verifikation verwendet wurde.

So wurden vier Mengen an Audiosignalen $\boldsymbol{A}_{\mathrm{SNR}}$ erzeugt:

$\boldsymbol{A}_{50\,\mathrm{dB}}$ enthält $3 \cdot 6 = 18$ Audiosignale, wobei jedes der sechs Baby-Aufnahmen mit jedem der drei Rauschsignale bei einem Signal/Rausch-Abstand von 50 dB überlagert wurde.

[2] Es wurden *keine* Geräusche markiert, bei denen es sich offensichtlich um Einatmungs-Geräusche handelt. Geräusche, bei denen nur Anhand der Aufnahme nicht mit Sicherheit festgestellt werden konnte, ob es sich um Einatmungs- oder Ausatmungsgeräusche handelt, wurden als Stimme markiert. Die Begründung liegt darin, dass nach Varallyay [51] bei der Einatmung entstehenden Geräusche keine verwertbare Information enthalten.

$\boldsymbol{A}_{3\,\mathrm{dB}}$ enthält $3 \cdot 6 = 18$ Audiosignale, wobei jedes der sechs Baby-Aufnahmen mit jedem der drei Rauschsignale bei einem Signal/Rausch-Abstand von 3 dB überlagert wurde.

$\boldsymbol{A}_{\mathbf{50}+3\,\mathrm{dB}} = \{A_{50\,\mathrm{dB}} \cup A_{3\,\mathrm{dB}}\} = 32$ Audiosignale

$\boldsymbol{A}_{7\,\mathrm{dB}*}$ enthält ein Audiosignal, bei dem eine siebte Aufnahme eines Babys mit einem vierten Rauschsignal bei einem Signal/Rausch-Abstand von 7 dB überlagert wurde

Aus diesen Audiosignalen wurden die tatsächlichen Datensätze $\boldsymbol{D}_{\mathrm{SNR,Feats}}$ nach dem folgenden Vorgehen erzeugt:

1. Jedes der Signale wurde nach dein Kapitel 4.1.1 vorgestellten Methoden vorverarbeitet und in Signalfenster à 25 ms zerlegt.
2. Jedes Signalfenster $x_i[\,]$ wurde zu ein Example e_i gewandelt, in dem eine bestimmte Untermenge der vorgestellten Features für das Signalfenster berechnet wurde. Tabelle 4.1 gibt eine Übersicht über die insgesamt 9 Untermengen. Die ersten vier Untermengen beinhalten jeweils die Features, die sich durch die jeweilige Methoden erzeugen lassen. Die nächsten fünf Untermengen enthalten jede mögliche paarweise Kombination der ersten vier Untermengen, mit Ausnahme des Cepstrum+Autokorrelation, da die Features dieser Bereiche am rechenaufwendigsten sind. Jedem Example wurde das entsprechende Klassenlabel beigefügt.

Tabelle 4.1: Übersicht über die gebildeten Feature-Untermengen

Name	verwendete Features
Zeit	RMS, Diff(RMS), ZCR, Diff(-ZCR)
Spektrum	SEnt_u, Diff(SEnt_u), SEnt_n, Diff(-SEnt_n), f_{dom}, Diff(f_{dom})
Autokorr.	aMax, Diff(aMax), aCount, Diff(-aCount)
Cepstrum	Ceps_{mag}, Diff(Ceps_{mag}), Ceps_{loc}
Zeit+Spektrum	RMS, ..., SEnt_u, ...
Zeit+Autokorr.	RMS, ..., aMax, ...
Zeit+Cepstrum	RMS, ..., Ceps_{mag}, ...
Spek.+Autokorr.	SEnt_u, ..., aMax , ...
Spek.+Cepstrum	SEnt_u, ..., Ceps_{mag} ,...

3. Alle Examples, die aus der selben Signalmenge $\boldsymbol{A}_{\mathrm{SNR}}$ stammen und die selbe Feature-Untermenge teilen, werden in einem Datensatz $\boldsymbol{D}_{\mathrm{SNR,Feats}}$ zusammengefasst. Beispielsweise enthält der Datensatz $\boldsymbol{D}_{50\,\mathrm{dB,Zeit}}$ alle Examples, die auf Grundlage der Signalmenge mit einem Signal/Rausch-Abstand von 50 dB unter Verwendung der in Tabelle 4.1 aufgelisteten Features des Zeitbereiches erzeugt wurden.
4. Da in allen Datensätzen rund dreimal mehr Positives (das heisst, Examples aus stimmhaften Signalfenstern) enthalten waren als Negatives, wurde jedes in einem Datensatz enthaltene Negative dreimal eingefügt. So wurde ein ausgewogenes Verhältnis an Positives und Negatives gewährleistet.

Auf diese Art und Weise wurden aus den vier Signalmengen $\boldsymbol{A}_{50\,\mathrm{dB}}, \ldots, \boldsymbol{A}_{7\,\mathrm{dB}*}$ und den 9 Feature-Untermengen insgesamt $4 \cdot 9 = 36$ *Trainingsdatensätze* $\boldsymbol{D}_{\mathrm{SNR,Feats}}$ gebildet. Wird einer dieser Datensätze als Testdatensatz verwendet, so sind die Features des Datensatzes unerheblich, da nur die Informationen der Klassenlabels beim Testing benötigt werden. Ein Testdatensatz $\boldsymbol{D}_{\mathrm{SNR}}$ kann also erzeugt werden, in dem bei einem beliebi-

gen Trainingsdatensatz mit dem Entsprechenden SNR die Featureinformationen ignoriert werden.

Training und Evaluation

Das Ziel ist es nun, auf Basis der Datensätze durch den *C4.5*-Algorithmus denjenigen Klassifikator zu finden, der für sowohl niedrige als auch hohe SNRs eine möglichst hohe Klassifikationsgenauigkeit erzielt. Es ist beispielsweise denkbar, dass ein Klassifikator, welcher auf Basis eines Datensatzes mit niedrigem SNR erzeugt wurde, sich nicht gut zur Klassifizierung hoher SNRs eignet, jedoch ein auf Basis niedriger SNRs entworfener Klassifikator ebenfalls eine zufriedenstellende Genauigkeit bezüglich hoher SNRs gewährleistet.

Um systematisch den besten Entscheidungsbaum zu erzielen, wurde auf Basis jedes Trainingsdatensatzes, mit Ausnahme der $D_{7\,\mathrm{dB}*}$-Datensätze, durch den *C4.5*-Algorithmus ein Klassifikator erzeugt. So wurden insgesamt $3 \cdot 9 = 27$ Klassifikatoren entworfen. Jeder Klassifikator wurde evaluiert, indem jeweils die Klassifikationsgenauigkeit für jeden der drei Testdatensätze $D_{3\,\mathrm{dB}}$, $D_{50\,\mathrm{dB}}$ und $D_{7\,\mathrm{dB}*}$ ermittelt wurde. Der $D_{7\,\mathrm{dB}*}$ erfüllt dabei eine Sonderfunktion, da er nicht zum Training verwendet wurde und somit der Kontrolle bezüglich Overfitting dient. Der Stern bei der Bezeichnung des Datensatzes verdeutlicht diese Sonderrolle.

Die Implementierung, die für den *C4.5*-Algorithmus verwendet wurde, ist der *REPTree*[3] der Open Source Data-Mining-Bibliothek *Weka*[4]. Die Implementierung hat den Vorteil, dass die maximale Tiefe des Entscheidungsbaumes festlegbar ist und somit die Komplexität des Baumes begrenzt werden kann, um Overfitting zu vermeiden. Die maximale Tiefe des REPTree wurde auf 2 gesetzt.

Ergebnisse

Die Evaluationsergebnisse sind in Tabelle .1 zu sehen. Für jeden Trainingsdatensatz wird die Genauigkeit des jeweiligen Klassifikators für jeden der drei Testdatensätze angegeben. Außerdem wurde der Durchschnittswert aller drei Genauigkeiten berechnet.

Der Feauture-Bereich, welche zu den höchsten Klassifikationsgenauigkeiten führten, ist das Cepstrum. Das einzige Feature dieses Bereiches, welches in Klassifikationsbäume eingebaut wurde, war das Diff($\mathrm{Ceps}_{\mathrm{mag}}$)-Feature. Die Entscheidungsbäume, die mit dem Diff($\mathrm{Ceps}_{\mathrm{mag}}$)-Feature entworfen wurden, erreichten eine über die drei Testdatensätze gemittelte Genauigkeit von mindestens 91,45%. Der nächstbeste Klassifikator mit einer gemittelten Accuracy von 86,96% wurde unter Verwendung der Features des Zeitbereiches und der Autokorrelation auf dem Datensatz $D_{50+3\,\mathrm{dB}}$ entworfen. Sobald das Cepstrum in Verbindung mit den Features anderer Bereiche verwendet wurden, wurde das Diff($\mathrm{Ceps}_{\mathrm{mag}}$)-Feature vom *C4.5*-Algorithmus bevorzugt und die Features der anderen Bereiche nicht mehr in die Entscheidungsbäume mit eingebaut.

Auf Basis der Datensätze $\mathrm{D}_{3\,\mathrm{dB,Ceps}}$, $\mathrm{D}_{3\,\mathrm{dB,Zeit+Ceps}}$, $\mathrm{D}_{3\,\mathrm{dB,Freq+Ceps}}$, $\mathrm{D}_{50+3\,\mathrm{dB,Ceps}}$, $\mathrm{D}_{50+3\,\mathrm{dB,Zeit+Ceps}}$ sowie $\mathrm{D}_{50+3\,\mathrm{dB,Freq+Ceps}}$ wurde der selbe Klassifikator erzeugt, der in Gleichung 4.19 gezeigt wird. Wie zu sehen ist, handelt es sich um einen einfachen Grenzwert

[3] Dokumentation von REPTree: `http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html`

[4] Download von WEKA: `http://www.cs.waikato.ac.nz/ml/weka/`

des *Diff(Ceps$_{mag}$)*-Features, da trotz der höchst möglichen Baumtiefe von 2 nur eine Tiefe von 1 genutzt wurde.

$$C(x_i[\,]) = \begin{cases} 1, & \text{wenn } Diff(Ceps_{mag}(c_i[\,])) > 0.02, \\ 0 & \text{sonst} \end{cases} \tag{4.19}$$

Auf Basis der Datensätze $D_{50\,dB,Ceps}$ und $D_{50\,dB,Zeit+Ceps}$ wurde der Klassifikator nach Gleichung 4.20 erzeugt. Er unterscheidet sich von dem Klassifikator aus Gleichung 4.19 nur durch die Höhe des Grenzwertes.

$$C(x_i[\,]) = \begin{cases} 1, & \text{wenn } Diff(Ceps_{mag}(c_i[\,])) > 0.03, \\ 0 & \text{sonst} \end{cases} \tag{4.20}$$

Da der Klassifikator aus Gleichung 4.19 eine durchschnittliche Genaugikeit von 92,22% und der Klassifikator aus Gleichung 4.20 eine unwesentlich geringere Genauigkeit von 91,45% erzielte, wurden für beide Modelle die Spezifität und Sensitivität berechnet, um eine Entscheidung für eines der beiden Modelle fällen zu können. Dazu wurden die Signalmengen $A_{3\,dB}$, $A_{50\,dB}$ und $A_{7\,dB*}$ in Frames à 100 Windows zerlegt und für jedes Zeitfenster die Sensitivität, Spezifität und Genauigkeit bezüglich der beiden Klassifikatoren berechnet. Die Ergebnisse werden als Boxplots in Abbildung .1 dargestellt. Die Modelle unterscheiden sich am stärksten hinsichtlich der Datensätze mit 3 dB und 7 dB. Der Klassifikator mit dem Grenzwert von 0.03 erzielt in beiden Fällen eine höhere Spezifität, aber geringere Sensitivität als das Modell mit dem Grenzwert von 0.02.

Es wurde sich für das Modell für mit einem Grenzwert von 0.02 entschieden, da durch die höhere Sensitivtät mehr Cry-Units erkannt werden, die in späteren Verarbeitungsschritten immernoch als False-Positives erkannt und verworfen werden können. Einmal im Prozess der VAD als Stimmlos markierte Fenster werden jedoch nicht weiter verarbeitet und gehen somit „verloren“. Die finale Klassifizierungsfunktion eines Signalfensters zur Voice Activity Detection ist somit die, die in Gleichung 4.19 abgebildet wird.

4.2 Markierung der Cry-Units

In Kapitel 4.1 wurde die Voice Activity Detection besprochen. Das Ergebnis der Untersuchung war eine Klassifizierungsfunktion, die ein Signalfenster als *stimmhaft* oder *nicht stimmhaft* markiert. Varallyay [51, S. 16 - 17] stellt die Idee vor, auf Grundlage des Ergebnisses der Voice Activity Detection die Anfangs- und Endzeitpunkte der Cry-Units zu markieren.[5] Das genaue Vorgehen konnte jedoch nicht eingesehen werden, da der Autor keine Zugriffsrechte auf die Publikation erhielt.

Waheed et al [24] stellen die Idee vor, zusammenhängende und ununterbrochene Ketten als *stimmhaft* klassifizierter Signalfenster zu *Stimm-Segmenten* zusammenzufassen. Dieser Ansatz wird übernommen, wobei ein Stimmsegment im Kontext dieser Arbeit einer *Cry-Units* entspricht. Möglicherweise ist dies der Ansatz, den auch Varallyay [51, S. 16 - 17] gewählt hat. Abbildung 4.10 veranschaulicht diese Gruppierung.

[5] „Cry-Units “ werden von Varallyay als „Cry-Segmente “ beschreibt.

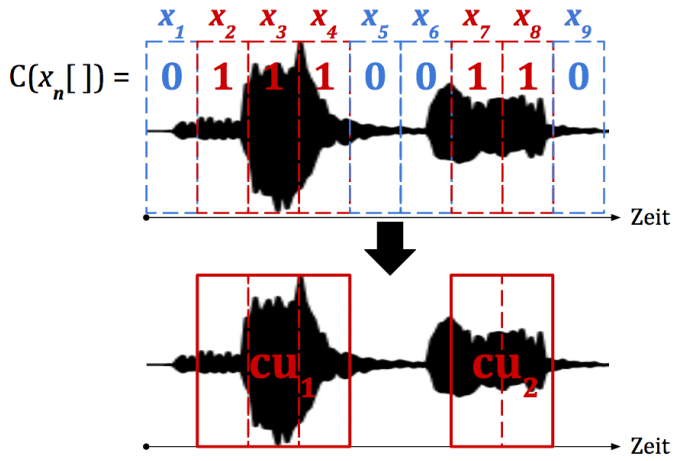

Abbildung 4.10: Zusammenfassung klassifizierter Signalfenster zu Cry-Units

Formel 4.21 gibt die Definition des Datentypes *Cry-Unit* $[CU]$. Eine Cry-Unit wird definiert durch den Anfangszeitpunkt *start*, einen Endzeitpunkt *end* und der Liste seiner Signalfenster $windows = [x_0[\,], \ldots, x_n][\,]$.

$$CU = (windows = [x_0[\,], \ldots, x_n[\,]], start \in Zeit, end \in Zeit) \tag{4.21}$$

Die Dauer eine Cry-Unit $cu \in CU$ wird nach Formel 4.22 berechnet und mit λ bezeichnet. Die zeitliche Dauer der Pause zwischen zwei Cry-Units $\mathrm{d}(cu_i, cu_j)$, wird nach Formel 4.23 berechnet. Diese Zusammenhänge werden in Abbildung 4.11 visualisiert.[24, S. 2]

$$\lambda(cu) = cu.end - cu.start \tag{4.22}$$

$$\mathrm{d}(cu_i, cu_j) = cu_j.start - cu_i.end \tag{4.23}$$

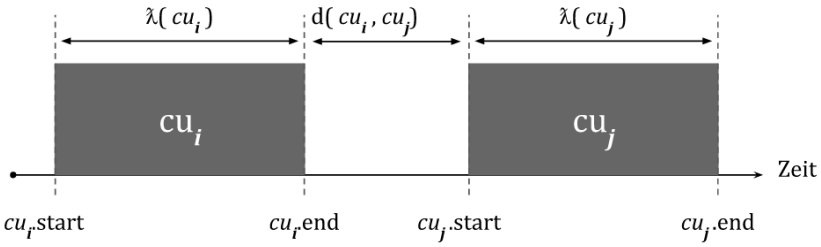

Abbildung 4.11: Beziehung zwischen agrenzenden Cry-Units, nach [24, S. 2]

Algorithmus 1 zeigt in Pseudo-Code, wie auf Basis der Liste aller Signalfenster eines Signals $X_{all} = [x_0[\,], \ldots, x_m[\,]]$ die Liste der Cry-Units CU_{all} generiert wird. Die Funktion $C(x[\,])$ ist die Klassifikations-Funktion der Signalfenster in Stille/Stimme nach Gleichung **??**. Die Funktion getTimeOf($x_i[\,]$) liefert die Anfangszeitpunkt des Signalfensters $x_i[\,]$.

Algorithm 1 Gruppierung von Signalfenstern zu Cry-Units

```
1:  function turnWindowsIntoCryUnits(X_all)
2:      CU_all ← [ ]
3:      cu ← ([ ], 0, 0)
4:      for all x_i[ ] ∈ X_all do
5:          c ← C(x_i[ ])
6:                                                  ▷ Start of Cry-Unit
7:          if c == 1 ∧ isEmpty(cu.windows) then
8:              cu ← ([ ], 0, 0)
9:              cu.start ← getTimeOf(x_i[ ])
10:             cu.windows ← [cu.windows, x_i[ ]]
11:         end if
12:                                                 ▷ Inside Cry-Unit
13:         if c == 1 ∧ ! isEmpty(cu.windows) then
14:             cu.windows ← [cu.windows, x_i[ ]]
15:         end if
16:                                                 ▷ End of Cry-Unit
17:         if c == 0 ∧ ! isEmpty(cu.windows) then
18:             cu.end ← getTimeOf(x_i[ ])
19:             CU ← [CU, cu]
20:             cu.windows ← [ ]
21:         end if
22:     end for
23:                                  ▷ End last Cry-Unit by force if still open.
24:     if ! isEmpty(cu.windows) == 0 then
25:         cu.end ← getTimeOf(X_windows[end])
26:         CU_all ← [CU_all, cu]
27:     end if
        return CU_all
28: end function
```

4.3 Decision Smoothing

Abbildung 4.12 zeigt ein Audiosignal mit einem Signal-Rausch-Abstand von 3 dB, bei dem die Voice Activity Detection durchgeführt wurde. Die rote Linie zeigt die tatsächliche Klassifiikation und die grüne Linie die prognostizierte Klassifikation. Es ist zu sehen, dass einige False-Negatives und prongnostiziert wurden. Im folgenden werden drei charakteristische Arten falscher Klassifikationen näher erläutert:

False Negatives nach (a): Eine korrekt erkannte, längere Cry-Unit wird zu früh beendet. Oft werden kurz nach dem Ende einer längeren Cry-Unit sehr kurze Cry-Units erkannt, die eigentlich noch zu der längeren, vorhergehenden Cry-Unit gehören.

False Positives nach (b): Kurze Cry-Units werden in eigentlichen Stille-Bereichen erkannt.

False Negatives nach (c): Eine Cry-Unit zerfällt in zwei Cry-Units, da kurze Signalfenster in der Mitte als Stille erkannt wurden.

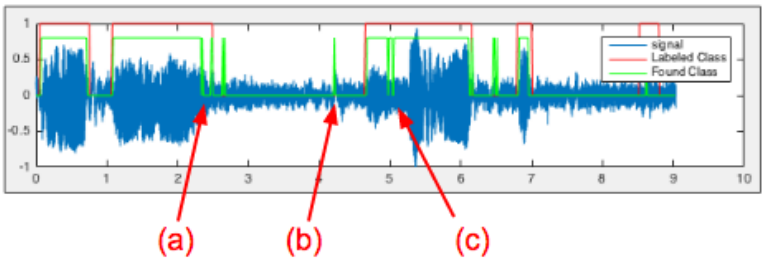

Abbildung 4.12: Klassifizierung vor dem Decision Smoothing

Im Process des **Decision Smoothing** werden kontextuelle Informationen genutzt, um nachträglich False-Positives und False-Negatives zu entfernen. Es werden dazu die von Waheed et al [24] präsentierten Ideen verwendet. Es werden zwei Parameter eingeführt: λ_{min}, die Mindestlänge einer akzeptierten Cry-Unit, und d_{min}, die Mindestlänge eines akzeptierten Stille-Segmentes. Das Decision-Smoothing wird nach den folgenden Entscheidungsregeln durchgeführt:

- ist $\lambda(cu_i) \leq \lambda_{min}$?
  - wenn $\lambda(cu_{i-1}) > \lambda_{min}$ und $d(cu_{i-1}, cu_i) \leq d_{min}$, dann vereinige cu_i mit cu_{i-1} . $\Longrightarrow$ behebt False-Negatives des Types (a)
  - ansonsten entferne $cu_i \Longrightarrow$ behebt False-Negatives des Types (b)
- wenn $\lambda(cu_i) > \lambda_{min}$ und $d(cu_{i-1}, cu_i) \leq d_{min}$, dann vereinige cu_i mit cu_{i-1} . $\Rightarrow$ behebt False-Negatives des Types (c)

Die Entscheidungsregeln greifen nur auf die letzten beiden erkannten Cry-Units zu, um eine kontinuierliche Analyse zu gewährleisten. Bei einer kontinuierlichen Analyse wird die Auswertung um die Zeitdauer einer Cry-Unit verzögert, da die Entscheidungsregeln erst nach Beendigung einer Cry-Unit abgefragt werden können. Bei einer offline-Analyse können die Entscheidungsregeln vereinfacht werden, da die False-Negatives nach Typ (a) und (c) mit der selben Regel abgefragt werden können. Algorithmus 2 zeigt in Pseudo-Code, wie das Decision-Smoothing durchgeführt wird. Input der Funktion ist die Liste aller Cry-Units $CU_{all} = [cu_0, \ldots, cu_n]$, die durch Algorithmus 1 entstanden ist, sowie die Grenzwerte λ_{min}, d_{min}. Der Output der Funktion ist die Liste aller Cry-Units nach dem Decision-Smoothing $CU_{smoothed}$.

Abbildung 4.13 zeigt das Beispielsignal vor und nach dem Decision-Smoothing. In verschiedenen Veröffentlichungen wurden unterschiedliche Mindestlängen von Cry-Units festgestellt. Varallyay [51, S. 8] hat beispielsweise eine Mindestlänge von 250 ms gemessen. Der geringste Wert, der nach dem Wissen des Autors in einer Veröffentlichung genannt wurde, stammt von Zeskind et al [40, S. 325] und beträgt 60 ms, welcher für λ_{min} übernommen wurde. Es konnten hingegen keine Werte über die geringste festgestellte Pause zwischen zwei Cry-Units gefunden werden. Der Wert wurde daher auf Basis des verwendeten Trainings-Datensatzes ebenfalls mit $d_{min} = 60$ ms bestimmt.

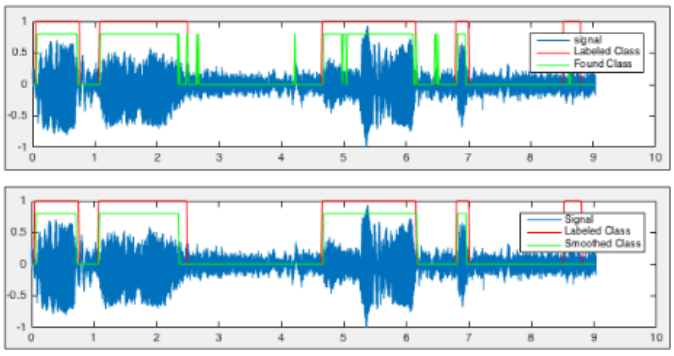

Abbildung 4.13: Klassifikation vor und nach dem Decision Smoothing

4.3.1 Diskussion der Voice-Activity-Detection

In diesem Kapitel wurden verschiedene Methoden der Voice Activity Detection vorgestellt, verglichen und evaluiert, wobei eine Voice Activity Detection auf Basis des Cepstrums die besten Ergebnisse erzielte. Unabhängig von den konkret verglichenen Features werden in

Algorithm 2 Decision-Smoothing for VAD

```
1: function DECISIONSMOOTHING(CU_all, λ_min, d_min)
2:     CU_smoothed ← [CU_all[0]]
3:                                   ▷ start for-loop at the second cry-Unit!
4:     for i = 1, ..., length(CU_all) − 1 do
5:         cu_i ← CU_all[i]
6:         cu_{i−1} ← CU_smoothed[end]
7:         if λ(cu_i) > λ_min then
8:                                                          ▷ Accept Cry-Unit
9:             if d(cu_{i−1}, cu_i) > d_min then
10:                CU_smoothed ← [CU_smoothed, cu_i]
11:            else
12:                                          ▷ Erase False-Negative Type (c)
13:                cu_i ← vereinige(cu_i, cu_{i−1})
14:                CU_smoothed ← [CU_smoothed[1 : end − 1], cu_i]
15:            end if
16:        else
17:                                          ▷ Erase False-Negative Type (a)
18:            if d(cu_{i−1}, cu_i) ≤ d_min then
19:                cu_i ← vereinige(cu_i, cu_{i−1})
20:                CU_smoothed ← [CU_smoothed[0 : end − 1], cu_i]
21:            else
22:                                  ▷ Don't accept cu_i. Erases False-Positives (b)
23:            end if
24:        end if
25:    end for
       return CU_smoothed
26: end function
```

dieser grundlegenden Herangehensweise zur Voice Activity Detection kontextuelle Informationen in Bezug auf den zeitlichen Verlauf der Stimme jedoch nur in einem geringen Maße beim Decision-Smoothing verwertet. Schlussendlich markiert der VAD-Algorithmus eine Reihe von kurzen Signalfenstern genau dann als zusammenhängende Cry-Unit, wenn jedes Signalfenster für sich betrachtet als Lautäußerung eines Babies klassifiziert wurde. Ob jedoch die Reihenfolge der in den Signalfenstern enthaltenen Lautäußerungen Sinn macht, wird nicht betrachtet. Schneidet man beispielsweise wenige Sekunden aus der Mitte einer längeren Cry-Unit aus und konkateniert dieses Sample viele Male, um eine synthetische, längere Cry-Unit zu erzeugen, klingt das Ergebnis für den Menschen stark unnatürlich, wird von dem hier vorgestellten VAD-Algorithmus jedoch trotzdem als valide Cry-Unit markiert. Das Cepstrum als Feature mit der höchsten Accuracy ist somit so zu bewerten, dass es vor allem im geringen Maße kontextuell Informationen benötigt, um eine Entscheidung über das vorahndensein von Stimme zu fällen. Zukünftige Forschungen könnnen an diesem Punkt ansetzen, um die Accuracy der VAD zu erhöhen.

5 Methoden zur Ableitung der Schmerz Score

5.1 Segmentierung

Das Ergebnis der Voice-Activiy-Detection ist eine Liste an Cry-Units $cu_0 \dots cu_n$. Pain-Scores werden nicht aus einzelnen Cry-Units abgeleitet, sondern aus dem Verbund mehrerer Cry-Units. Daher ist es notwendig, die Cry-Units zu Segmenten zusammenzufassen. Dieser Prozess des Gruppieren von Cry-Units zu Segmenten wird in dieser Arbeit kurz als *Segmentierung* (engl. *Segmenting*) bezeichnet. Die Frage ist, nach welchen Kriterien Cry-Units zu Segmenten zusammengefasst werden. Abbildung 5.1 verdeutlicht das Problem, in dem drei mögliche Segmentierungen für eine Signal beispielhaft gezeigt werden.

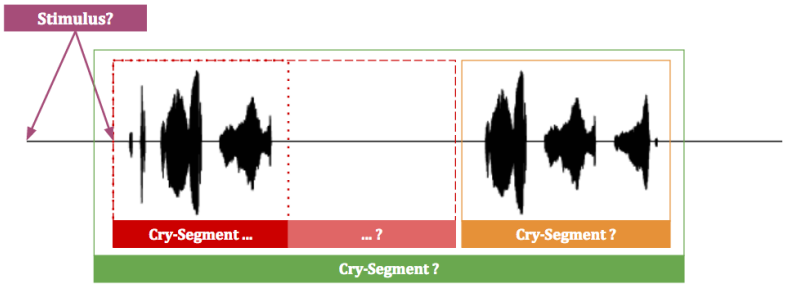

Abbildung 5.1: Mögliche Segmentierungen eines Signals

Ein Cry(-Segment) wird von Golub et al definiert als „die komplette klangliche Antwort auf einen spezifischen Stimulus. Sie kann mehrere Cry-Units enthalten “. [17, S. 61, übersetzt aus dem Englischen]. Die Defintion lässt unter Anderem die folgenden Fragen offen:

- Beginnt das Segment bereits bei Zuführung des Stimulus, oder erst ab der ersten Cry-Unit?
- Wodurch definiert sich der Beginn, wenn der Stimulus unbekannt ist?
- Endet ein Cry-Segment mit Ende der letzten „Cry-Unit“, oder erstreckt es sich bis zu Beginn des nächsten Cry-Segmentes?

Keines der in Kapitel 3.1 vorgestellten Veröffentlichungen schlägt Methoden zur Segmentierung vor. Bei den nicht-kontinuierlichen Systemen werden manuell beschnittene Cry-Segmente verwendet. Entweder werden keine objektiv messbaren Krtierien gegeben (außer “das Segment dort zu beenden, wo das Baby aufhört, zu weinen“), oder feste Längen wie zum Beispiel 90 s[40, S. 324] gegeben. Bei den kontinuierlichen Systemen wird die Segmentierung nicht als Verarbeitungsschritt erwähnt, eventuell, weil keine stattfindet.

Es wird daher das folgende Vorgehen zur kontinuierlichen Segmentierung vorgeschlagen: Wenn das Baby keine Äußerungen von sich gibt, weil es beispielsweise schläft, wird keine

Cry-Unit festgestellt, und somit existiert auch momentan kein offenes Segment. Fängt das Baby an, einen Laut von sich zu geben, also eine Cry-Unit zu produzieren, wird ein neues Segment eröffnet und die Cry-Unit diesem Segment hinzugefügt. Weitere Cry-Units werden so lange diesem Segment hinzugefügt, wie die Dauer der Stille nach einer Cry-Unit einen festgelegten Grenzwert t_s nicht überschreitet. Ein Cry-Segment wird folglich dann geschlossen, wenn das Baby „aufhört, zu weinen“, also keine Laute mehr für einen festgelegten Zeitraum von sich gibt. Das Endzeitpunkt des Segmentes wird als der Endzeitpunkt der letzten Cry-Unit des Segmentes festgelegt.

Formel 5.1 definiert ein *Cry-Segment* $[CS]$ als Datentyp. Ein Cry-Segment ist eine Liste von Cry-Units. Alle Cry-Units erfüllen die Nebenbedingung 5.2, das heißt, dass die Distanzen aller benachbarter Cry-Units eines Cry-Segments unterhalb des Grenzwertes t_s liegen.

$$CS = [cu_0, \ldots, cu_n] \tag{5.1}$$

$$\forall cs \in CS : \forall i = 0 \ldots length(cs) - 2 : d(cs[i], cs[i+1]) < t_s \tag{5.2}$$

Der Start-Zeitpunkt eines Cry-Segmentes wird nach Formel 5.3 als der Startzeitpunkt der ersten Cry-Unit des Segmentes definiert. Das Ende eines Segmentes wird definiert als das Ende der letzten Cry-Unit nach Gleichung 5.4.

$$start(cs) = cs[0].start \tag{5.3}$$

$$end(cs) = cs[n].end \tag{5.4}$$

Algorithmus 3 zeigt einen Pseudocode, wie die Segmentierung nach dem beschriebenen Prinzipien offline durchgeführt wird. Input des Algorithmus ist die Liste aller Cry-Units $CU_{all} = [cu_0, \ldots, cu_m]$, die nach dem Decision-Smoothing nach Algorithmus 2 entstanden ist. Das Ergebnis des Algorithmus ist die Liste, die alle gefundene Cry-Segmente $[cs_0 \ldots cs_n]$ enthält. Der Algorithmus eignet sich nicht für eine Online-Segmentierung, da das Ende eines Segmentes erst nach dem Abschluss einer Cry-Unit festgestellt wird, wobei beliebig viel Zeit zwischen zwei Cry-Units liegen kann. Bei einer online durchgeführten Segmentierung empfiehlt es sich, ein Segment sofort zu beenden, wenn der Zeitraum der Stille nach einem Segment den Grenzwert t_s überschreitet.

Algorithm 3 Gruppierung von Cry-Units zu Cry-Segments

function SEGMENTCRYUNITS(CU_{all}, t_s)
  $CS_{all} \leftarrow [\,]$
  $cs \leftarrow [CU_{all}[0]]$
  for $i = 1, \ldots, length(CU_{all}) - 1$ **do**
    $cu_i \leftarrow CU_{all}[i]$
    $cu_{i-1} \leftarrow CU_{all}[i-1]$
    if $\mathrm{d}(cu_{i-1}, cu_i) < t_{seg-max}$ **then**
      $cs \leftarrow [cs_i, cu_i]$
    else
      $CS_{all} \leftarrow [CS_{all}, cs]$
      $cs \leftarrow [cu_i]$
    end if
  end forreturn CS_{all}
end function

Abbildung 5.2 zeigt die nach dieser Methode durchgeführte Segmentierung anhand eines Beispiels.

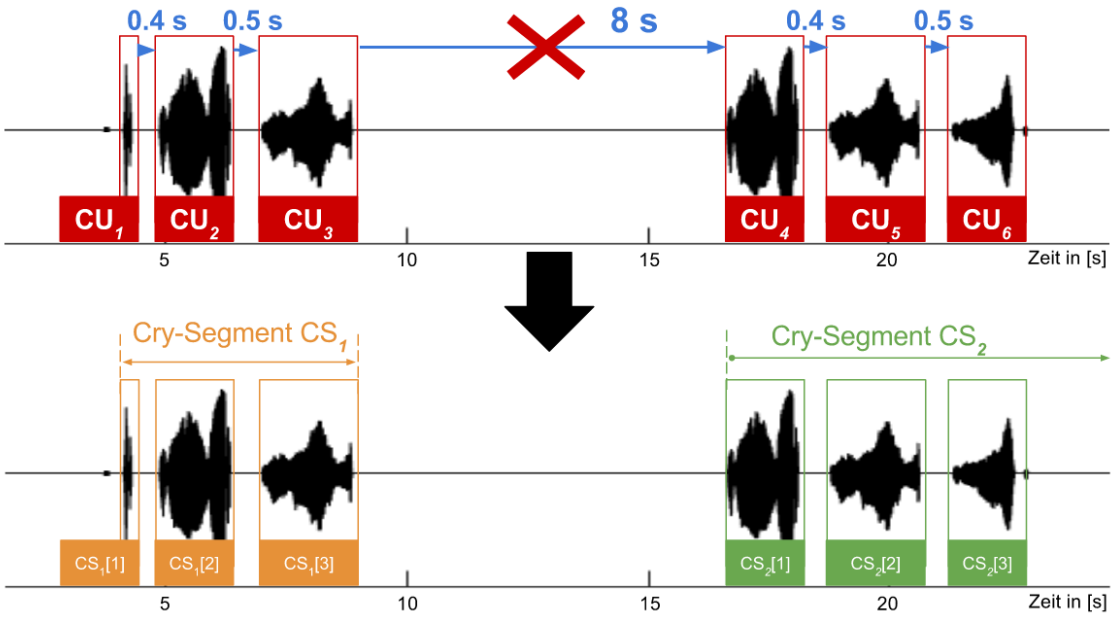

Abbildung 5.2: Ergebnis der Segmentierung mit einem Grenzwert von $t_s = 6\,\mathrm{s}$

Das hier vorgestellte Vorgehen ist absichtlich möglichst einfach gehalten, damit der Sinn des Parameters t_s leicht ersichtlich ist und somit von der medizinischen Fachkracht selbständig festgelegt werden kann. Schlussendlich ist eines der Hauptziele dieser Segmentierung, unnötige Berechnungen von Schmerz-Scores in den nachfolgenden Schritten zu vermeiden, so lange keine Cry-Units vorliegen. Das Ende eines Segmentes ist außerdem ein günstiger Zeitpunkt, um die Parameter des Kompressors im Pre-Processing auf Basis des RMS-Wertes des Segmentes zu aktualisieren (siehe Kapitel 4.1.1). Trotz der Trivialität dieser laufenden Segmentierung liegt hier ein wichtiger Unterschied im Gegensatz zu vergleichbaren Systemen,

wie zum Beispiel das von Cohen et al [6], bei dem die Entscheidung über Cry/not-Cry für Segmente mit einer festen Fenstergröße von 10 Sekunden vorgenommen wird.

5.2 Feature-Extraktion und Ableitung der Schmerzscore

Das Erebnis der Segmentierung ist eine Litse an Cry-Segmenten $cs_0, \ldots, c_n$. Diese Cry-Segmente bilden nun die Basis für die Ableitung der Pain-Score[1]. Die medizinische Fachkraft, die das System verwendet, muss dabei zuerst die Wahl treffen, welche Pain-Scale verwendet werden soll. Das einfachste denkbare Vorgehen ist die Ableitung genau einer Pain-Score aus den globalen Eigenschaften eines Segmentes, wobei diese Ableitung erst vollzogen werden kann, sobald ein Segment abgeschlossen wurde und alle Informationen für dieses Segment vorliegen. Es wird also jedem Segment genau eine Pain-Score zugewiesen. Das Vorgehen wird am Beispiel der NIPS aus Tabelle 2.1 verdeutlicht: Dabei steht die Abwesenheit von Weinen für null Punkte, „mumbling“ (murmeln) für einen Punkt und „vigorous“ (energisch) für zwei Punkte. Bei Abwesenheit von Lautäußerungen, also der Zeitraum zwischen den Segmenten, werden also keine Punkte = null Punkte vergeben. Ein Segment, dessen Qualität insgesamt als „murmelnd“ bewertet wird, erhält einen Punkt, und ein Segment, welches als insgesamt als „energisch“ bewertet wird, zwei Punkte. Das Problem ist offensichtlich: „murmelnd“ und „energisch“ sind subjektiv behaftete Begriffe und lassen sich nicht ohne weiteres aus den Eigenschaften eines Segmentes feststellen.

Es werden zwei verschiedene Lösungs-Strategien für dieses Problem vorgestellt.

Strategie 1

... löst das Problem mit Hilfe von *Regression* (Siehe Kapitel **??**):

1. Man erstellt eine Datenbank mit Aufnahmen von kindlichen Lautäußerungen, die man segmentiert.
2. Man errechnet „so viele *objektiv* messabare Eigenschaften wie möglich“ für jedes Segment, wie zum Beispiel die insgesamte Länge, die durchschnittliche Länge der enthaltenen Cry-Units, durchschnittliche Tonhöhe usw.
3. Man bittet medizinische Fachkräfte, für jedes Segment der Datenbank eine Score bezüglich einer Pain-Scale zu vergeben. Dadurch erhält man eine gelabelte Test-Datenbank.
4. Man verwendet einen *Regressionsalgorithmus*, um den Zusammenhang zwischen den in Schritt 2 objektiv gemessenen Eigenschaften der Segmente und den in Schritt 3 vergebenen *Scores* herzustellen. An dieser Stelle kann zum Beispiel die in Kapitel **??** beschriebene multiple lineare Regression verwendet werden. Man erhält somit einen Regressor für jede Pain-Scale.
5. Möchte man für neue, unbekannte Segmente die Pain-Score ableiten, nutzt man den entsprechenden Regressor.

[1] Um Unklarheiten zu vermeiden, wird an dieser Stelle noch einmal darauf hingewiesen, dass mit „Pain-Scale“ eine Scale, wie zum Beispiel die NIPS gemeint ist, und mit „Pain-Score„ oder einfach nur „Score“ die tatsächlich vergebene Punktzahl auf Basis der Bewertungskriterien der Pain-Scale

Das Vorteil dieses Vorgehens ist, dass das Problem der Übersetzung der objektiv messbaren Parameter in die subjektiv behafteten Begriffe überbrückt wird, indem die Regression direkt von den objektiv messbaren Parametern auf die Pain-Score durchgeführt wird. Der Nachteil ist, dass eine Testdatenbank für jede Pain-Scale aufgebaut werden muss. Wird ein neue Pain-Scale eingeführt, muss der Regressor für diese Scale durch erneutes Labeln festgestellt werden. Ein weiterer Effekt der Abbildung des Problems als Regression ist, dass ein Regressor in einen kontinuierlichen Zahlenraum abbildet. Es sind also Regressionsergebnisse wie zum Beispiel 2.8 denkbar. Diese „bessere Auflösung“ kann als Vorteil betrachtet werden. Ist jedoch eine direkte Übersetzung der Pain-Scale inklusive der ganzzahligen Punktzahlen gewünscht, so stellt sich die Frage, ob eine 2.8 auf- oder abzurunden ist.

Strategie 2

... löst das Problem mit Hilfe von Klassifizierung (Siehe Kapitel **??**):

1. und 2. entsprechen Strategie 1
3. Man sammelt alle subektiven Begriffe, die in Pain-Scales verwendet werden, wie zum Beispiel „murmelnd“, „energisch“, usw.
4. Man bittet medizinische Fachkräfte, jedes Segment der Datenbann mit denjenigen Begriffen zu labeln, die die jeweilige Person für zutreffend hält.
5. Man Verwendet einen *Klassifizierungsgorithmus*, um einen Zusammenhang zwischen den in Schritt 2 festgestellten objektiv messbaren Eigenschaften der Segmente und den *subjektiv behafteten Begriffen* zu finden. Man erhält somit einen Klassifikator für jedenBegriff, der binär in *positive = zutreffend* und *negative = nicht zutreffend* klassifiziert.
6. Möchte man für neue, unbekannte Segmente die Pain-Score ableiten, so wird für jede Puntkzahl der Pain-Scale überprüft, ob für alle subjektiv beschreibenden Begriffe der entsprechende Klassifikator ein positive prognostiziert. Die Ableitung der Score ist somit ein weiters Klassifizierungsproblem, wobei eine Score einer Klasse entspricht und genau dann abgeleitet werden kann, wenn alle Vorraussetzungen für die Klasse erfüllt sind.

Der Vorteil dieser Methode ist, dass auch zum Zeitpunkt der Erstellung der Testdatenbank unbekannte Pain-Scales zu einem späteren Zeitpunkt eingebunden werden können, insofern alle in dieser neuen Pain-Scale verwendeten subjektiv behafteten Begriffe bereits gelabelt vorliegen, weil sie auch in anderen Pain-Scales verwendet werden. Das Vorgehen erlaubt somit eine gewissen Flexibilität bezüglich zukünftig entwickelter Pain-Scales. Der Nachteil dieser Methode ist, dass durch die Umwandlung der eigentlich quantitativ geordenten Score einer Pain-Scale in qualitative Klassen aus einem implizit als Regression zu betrachtenden Problem ein Klassifizierungsproblem macht. Dies wirft neue Fragen auf, wie zum Beispiel: Angenommen, bei einer fiktiven Pain-Scale wird jede Score mit jeweils drei subjektiv behafteten Begriffen beschrieben, und bei der Klassifizierung eines Segmentes wird festgestellt, dass für jede Punktzahl genau zwei der drei Begriffe erfüllt werden. Welche Score wird dann abeleitet? Ein anderes Beispiel wird am Beispiel der der NIPS-Score aus Tabelle 2.1 verdeutlicht: Angenommen, ein Cry-Segment enthält hörbar „starkes“ Schreien, es kann jedoch weder „mumbling (murmelnd) “ noch „vigorous (energisch)“ abgeleitet werden. Demzufolgen müsste dieses Segment eine Score von 0 Punkten erhalten, wobei ein Mensch

in dieser Situation eventuell „stark“ zu „heftig“ uminterpretieren und 2 Punkte vergeben würde. Strategie 1 ist weniger anfällig für dieses Problem.

In jedem Fall werden medizinische Fachkräfte benötigt, um das Labeling der Cry-Segmente durchzuführen, was aus Zeitgründen im Rahmen dieser Arbeit nicht möglich ist. Die Aquise von Audioaufnahmen von Babys sowie das Labeling der Aufanhmen erfodern nicht nur Zeit, sondern das Fachwissen über das Führen und die Auswerten von Interviews.

5.2.1 Extrahierung von Eigenschaften

Im vergangenen Kapitel wurde erläutert, dass die Basis für die Ableitung einer Pain-Score für ein Segment die Extraktion von „so vielen Features wie möglich“ ist. In diesem Kapitel wird präzisiert, welche Features gemeint sind. Varallyay [51, S. 16 - 17] schlägt vor, drei Kategorien an Features zu betrachten: (1.) dem Zeitbereich, (2.) dem Frequenzbereich, und (3.) Melodie-bezogene Attribute. Diese Kategorisierung wird übernommen.

In Kapitel 2.3.1 wurde beschrieben, welche Features in der medizinischen Schreiforschung typischerweise extrahiert werden. In Kapitel 2.3.2 wurde diskutiert, dass (1.) nicht bewiesen ist, welche Features die „wichtigsten“ sind und (2.) keine Einigung darüber herrscht, wie genau bestimmte Features zu berechnen sind. An dieser Stelle werden daher Berechnungsvorschriften für eine umfassende Auswahl an Features vorgestellt. Die Features basieren auf den Ideen, die in Kapitel 2.3.1 vorgestellt wurden, und erweitern diese logisch. Welche von diesen Features tatsächlich im Zusammenhang mit Schmerz stehen, lässt sich erst in der anschließenden Nutzung der Features zur Regression oder Klassifizierung der Pain-Scales feststellen, welche jedoch im Rahmen dieser Arbeit nicht durchgeführt werden kann.

Features des Zeitbereiches

Mit Features des Zeitbereiches sind solche gemeint, die sich allein aus Kenntnis der Cry-Units des Segments gewinnen lassen, wie beispielsweise die durchschnittliche Länge der Cry-Units, durchschnittliche Pause zwischen den Cry-Units, das relative Verhältnis von Cry-Units zu Pausen usw. Die folgenden Features werden konnkret definiert. In diesem Kapitel gilt die Konvention, dass eine Cry-Segment cs insgesamt N Cry-Units enthält, die Indexierung wird mit $0 \dots N-1$ definiert.

Segment-Length: Zeitliche Länge des Segmentes:

$$\text{Segment-Length}(cs) = cs[N-1].end - cs[0].start \tag{5.5}$$

Density: Relativer Anteil der Cry-Units an der Länge des Segmentes („Dichte“)

$$\text{Density}(cs) = \frac{\sum_{i=0}^{N-1} \lambda(cs[i])}{\text{Segment-Length}(cs)} \tag{5.6}$$

Tempo: Das Verhältnis zwischen der Dauer des Segmentes und der Anzahl der Cry-Units. Dieses Feature wird von LaGasse et al [27, S. 85] als *Utterances* bezeichnet.

$$\text{Tempo}(cs) = \frac{N}{\text{Segment-Length}(cs)} \tag{5.7}$$

Statistics of Cry-Units: Statistische Auswertungen bezüglich der *Länge der Cry-Units* $\text{stats}_{cu}(cs)$: Durchschnitt, Median, Minimum, Maximum und Standardabweichung der Cry-Units. Das $\text{mean}_{cu}(cs)$-Feature wird von LaGasse et al [27, S. 85] und vielen weiteren Schreiforschern als *Mean Duration* bezeichnet.

$$\text{stats}_{cu}(cs) = \begin{cases} \text{mean}_{cu}(cs) = \underset{i=0\ldots N-1}{\text{mean}}\{\lambda(cs[i])\} \\ \text{median}_{cu}(cs) = \underset{i=0\ldots N-1}{\text{median}}\{\lambda(cs[i])\} \\ \min_{cu}(cs) = \underset{i=0\ldots N-1}{\min}\{\lambda(cs[i])\} \\ \max_{cu}(cs) = \underset{i=0\ldots N-1}{\max}\{\lambda(cs[i])\} \\ \sigma_{cu}(cs) = \sigma_{i=0\ldots N-1}\{\lambda(cs[i])\} \end{cases} \tag{5.8}$$

Statistics of Bursts: [2] Die in Gleichung 5.8 definierten Features können ebenso in Bezug auf die *Längen der Bursts* errechnet werden, in dem in jeder Gleichung $\lambda(cs[i])$ ersetzt wird durch $cs[i].start - cs[i-1].start$. Die Indexierung muss auf $i = 1, \ldots, N-1$ begrenzt werden.

$$\text{stats}_{burst}(cs) = \begin{cases} \text{mean}_{burst}(cs) = \underset{i=1\ldots N-1}{\text{mean}}\{cs[i].start - cs[i-1].start\} \\ \text{median}_{burst}(cs) = \underset{i=1\ldots N-1}{\text{median}}\{cs[i].start - cs[i-1].start\} \\ \ldots \end{cases} \tag{5.9}$$

Statistics of Pauses: Nach dem selben Muster werden die statistischen Auswertungen bezüglich der *Längen der Pausen* ermittelt. Eine Pause entspricht in diesem Zusammenhang der Distanz zweier aufeineraderfolgenden Cry-Units, welche in Kapitel **??** definiert wurde.

$$\text{stats}_{pause}(cs) = \begin{cases} \text{mean}_{pause}(cs) = \underset{i=1\ldots N-1}{\text{mean}}\{d(cs[i-1], cs[i])\} \\ \text{median}_{pause}(cs) = \ldots \end{cases} \tag{5.10}$$

Statistics of Energies: Zunächst wird die Liste aller in den Cry-Units enthaltenen Signalfenster definiert nach Gleichung 5.11. Eine Cry-Unit hat die Signalfenster $cu.windows = x_0[\,], \ldots, x_m[\,]$

$$x_{seg}[\,] = cs[0].windows[0]\ ,\ \ldots\ ,\ cs[N-1].windows[m] \tag{5.11}$$

Die Liste $x_{seg}[\,]$ hat R Elemente, die Indexierung wird definiert mit $0, \ldots, R-1$. Gleichung 5.12 definiert die Features bezüglich der MSV-Werte („Lautstärken“) des Segmentes. Der MSV-Wert als Maß des durchschnittlichen Energiegehaltes wurde in Gleichung 2.6 definiert.

$$\text{stats}_{msv}(cs) = \begin{cases} \text{mean}_{E}(cs) = \underset{i=0\ldots R-1}{\text{mean}}\{MSV(x_{seg}[i])\} \\ \text{median}_{E}(cs) = \ldots \end{cases} \tag{5.12}$$

[2] Erläuterung zum Begriff *Burst* in 2.3.1)

Diese statistischen Auswertungen bezüglich der Länge der Cry-Units und Bursts wurden beispielsweise von Zeskind et al [40] vorgenommen, wenn auch nicht Computer-gestützt. Es ist zu bemerken, dass in der klassischen Schreiforschung zeitliche Features im geringeren Maße in Betracht gezogen wurden als Features des Frequenz-Bereiches. Die einzigen zeitliche Features, die zum Beispiel von Wasz-Hockert et al [37], Fuller [10] und LaGasse et al[27] in Betracht gezogen wurden, sind *die durchschnittliche Länge der Cry-Units* (hier $\text{mean}_{cu}(cs)$) und die *Latenz zwischen Reiz und erster Cry-Unit*, welche nur auf Basis des Audiosignals nicht feststellbar ist. Sie werden an dieser Stelle trotzdem berechnet, da nicht auszuschließen ist, dass sie zur Ableitung des Schmerzgrades eine Bedeutung erfüllen. Die anschließende Nutzung der Features zur Regression/Klassifizierung wird Auskunft darüber geben, welchen Beitrag diese Features zur Schmerzdiagnose leisten können.

Features des Frequenzbereiches und der Melodie

Mit Features des Frequenz-Bereiches sind diejenigen Features gemeint, die sich aus der Short Time Fourier Transformation der Cry-Units gewinnen lassen. Um die Features durch mathematische Formeln definieren zu können, wird zuerst das *Spectrum des Segmentes* $X_{seg}[\,]$ nach Formel 5.13 als die Liste aller Frequenz-Bereiche der Signalfenster der Cry-Units des Segmentes definiert. Die Indexierung von $X_{seg}[\,]$ läuft, wie bei $x_{seg}[\,]$ von $0, \ldots, R-1$. Nach dem selben Muster wird wird das *Cepstrum des Segmentes* $c_{seg}[\,]$ definiert.

$$X_{seg}[\,] := \underset{x_i[\,] \in x_{seg}}{\forall} : \; |DFT\{x_i[\,] \cdot w[\,]\}| \tag{5.13}$$

Die folgenden Features des Frequenzbereiches lassen sich mit den in dieser Arbeit vorgestellten Methoden berechnen:

Tensness: Das Feature, welches in Kapitel 2.3.1 als „Ratio2“ beschrieben wurde. Es wurde von Fuller [10] eingeführt und beschreibt die Spannung des Vokaltraktes als Verhältnis der Energien oberhalb von 2000 2000 Hz zu unter 2000 Hz . Wie bei den statistischen Auswertungen der Features des Zeitbereiches kann für das gesamte Segment der Durchschnitt, Median, Maximum, Minimum und Standardabweichung berechnet werden.

$$\text{stats}(Tensness) = \begin{cases} \text{mean}_{Tens}(cs) = \underset{i=0\ldots R-1}{\text{mean}} \left\{ \frac{\sum_{k=0}^{2000\,\text{Hz}} X_{sec}[i][k]}{\sum_{j=2000\,\text{Hz}}^{f_s} X_{sec}[i][j]} \right\} \\ \text{median}_{Tens}(cs) = \ldots \end{cases} \tag{5.14}$$

Clarity: Wie in Kapitel **??** erläutert wurde, lässt eine stark ausgebildete Spitze im oberen Cepstrum-Bereich auf ein stimmhaftes Signal schließen. Ein hoher Anteil stärkerer Cepstrum-Peaks lässt also auf vermehrt phonierte Laute schließen, geringere Cepstrum-Peaks auf dysphoniertere Laute (Siehe Kapitel 2.3.1). Dieses Feature trifft Aussagagen über den Anteil dysphonierter Laute, die Standardabweichung ähnelt dem in Kapitel 2.3.1 vorgestellten *Cry-Mode Changes*-Feature.

$$\text{stats}_{clarity}(cs) = \begin{cases} \text{mean}_{Clarity}(cs) = \underset{i=0\ldots R-1}{\text{mean}} \left\{ Ceps_{mag}(c_{seg}[i]) \right\} \\ \text{median}_{Clarity}(cs) = \ldots \end{cases} \tag{5.15}$$

Alle weiteren Features, die in Kapitel 2.3.1 vorgestellt wurden und sich auf den Frequenzbereich beziehen, lassen sich nicht mehr mit den in dieser Arbeit vorgestellten Methoden extrahieren. Entweder beziehen sie sich auf die Lage der Formanten, oder basieren auf der Feststellung der Grundtonhöhe. In dieser Arbeit konnten aus Platzgründen jedoch keine Methoden zur Extraktion dieser Informationen mehr vorgestellt. Gleiches gilt für die Feststellung des Melodieverlaufs, welche ebenfalls auf der Feststellung der Grundtonhöhe basiert. Das Muster, nach dem diese Features berechnet werden können, sollte aus den bisher vorgestellten Features ersichtlich sein. So lassen sich beispielsweise die Features bezüglich der Grundtonhöhe nach Formel 5.16 ableiten. Dabei sei $f_0(X_i[\,])$ eine idealisierte Funktion, welche die Grundtonhöhe f_0 für das Frequenzfenster $X_i[\,]$ berechnet. Da für die Definition der weiteren Features idealisierte ebenfalls Funktionen angenommen werden müssten, wird die Festlegung weiterer Features an dieser Stelle nicht fortgeführt.

$$\text{stats}_{pitch}(cs) = \begin{cases} \text{mean}_{Pitch}(cs) = \underset{i=0\ldots R-1}{\text{mean}} \left\{ f_0(X_{seg}[i]) \right\} \\ \text{median}_{Pitch}(cs) = \ldots \end{cases} \tag{5.16}$$

Diskussion

Bei allen vorgestellten Features handelt es sich, nach dem Vorbild der in Kapitel 2.3.1 vorgestellten Features der klassischen Schreiforschung, um solche, bei denen die Reihenfolge der Cry-Units nicht mit in Betracht gezogen wird. Angenommen, ein Segment besteht aus n Cry-Units, wobei genau eine hälfte der Cry-Units kurz und die andere hälfte der Cry-Units lang ist. Das $\text{stats}_{cu}(cs)$-Feature wird bezüglich des Durcschnittes, Minimum, Maximum etc. die selben Werte berechnen, unabhängig davon, ob sich die kurzen Cry-Units allesamt am Beginn des Segmentes, am Ende des Segmentes oder mit den langen Cry-Units durchmischt befinden. Bei der anschließenden Nutzung der Features zu Regression/Klassifizierung wird sich zeigen, wie sehr sich diese Features zur Ableitung von Pain-Scores eignen. Stellt sich heraus, dass sich die Features nicht eignen, ist es eventuell notwendig, die Position der Cry-Units in einer neuen Reihe von Features mit in Betracht zu ziehen.

5.2.2 Ableitung der Pain-Score

Zu Beginn von Kapitel 5.2 wurde gesagt, dass genau eine Score für ein Segment abgeleitet wird. Dies der einfachste denkbare Fall, welcher für einige Anwendungsfälle eventuell nicht ausreichend ist:

1. Kann die Score erst nach der Beendigung eines Segmentes abgeleitet werden, was in einigen Kontexten möglicherweise zu spät ist. So kann es notwendig sein, bereits eine Score abzuleiten, bevor das Segment beendet wurde, um zum Beispiel das schnelle Reagieren auf akuten und starken Schmerz zu ermöglichen.
2. Falls der Schmerz innerhalb eines Segmentes stark ab- oder zunimmt, ist dieser Verlauf nicht erkennbar. Es würde lediglich der „durchschnittliche Schmerz“ des Segmentes abgeleitet werden.

Das vorgestellte Prinzip wird daher erweitert, indem ein Aktualisierungsintervall t_{act} und Beobachtungszeitraume t_{obs} eingeführt wird.

Die Grundlegende Idee des Aktualsisierungsintervalles ist, bei einem momentan offenen

Segment in regelmäßigen Abständen die Features abzufragen und direkt die Pain-Score abzuleiten, um Zwischenergebnisse zu erhalten. Der am häufigsten umsetzbare Fall ist, ein Aktualisierung nach jeder neu dem Segment hinzgefügten Cry-Unit vorzunehmen. Der am wenigsten häufige Fall ist der bereits genannte, die Aktualisierung erst bei Beendingung eines Segmentes durchzuführen. An den in Kapitel 5.2.1 vorgestellten Formeln ändert dies nichts, wenn zum Aktualisierungszeitpunkt das Ende des Segmentes angenommen wird. Wird die Entscheidung über die Aktualisierungshäufigkeit der medizinischen Fachkraft überlassen, empfiehlt es sich, den Parameter möglichst einfach verstehbar zu machen, in dem man einen festen Intervall t_{act} festlegen lässt. Ein t_{act} von beispielsweise 10 s bedeutet, dass alle 10 Sekunden ein neuer Pain-Score für ein Segment berechnet wird. Die Beendigung eines Segmentes würde in jedem Fall eine Ableitung der Pain-Score auslösen und einen „erzwungenen Aktualisierungszeitpunkt“ darstellen. Es ist denkbar, das Aktualisierungsintervall fest an eine Pain-Scale zu binden. Die CRIES-Scale ist beispielsweise für das post-operative Monitoring gedacht und benötigt somit möglicherweise weniger häufige Aktualisierungen als der DAN, welcher zur Schmerzdiagnostik während einer Operation eingesetzt werden kann. [35, S. 98]

Die Idee hinter der Festlegung des Beobachtungszeitraumes ist die Verkürzung des Zeitraumes, der zur Feature-Berechnung verwendet wird. Es gibt Eigenschaften, die sich implizit auf den gesamten Zeitraum *Beginn des Segmentes* bis *Aktualisierungs-Zeitpunkt* beziehen, wie beispielsweise die *Zeitliche Länge des Segmentes* aus Formel 5.5. Dieser Zeitraum ist gleichzeitig der längst mögliche Zeitraum innerhalb eines Segmentes. Es ist jedoch auch möglich, kürzere Beobachtungszeiträume zu wählen. Dies hat zur Folge, dass die ersten Cry-Units des Segmentes ausgelassen werden, die außerhalb des Beobachtungszeitraumes liegen. Ist der Beobachtungszeitraum länger als die momentane Länge des Segmentes, werden die Berechnungen für das gesamte Segment durchgeführt. So können zeitliche Veränderungen der Pain-Score innerhalb eines Segmentes detaillierter dargestellt werden. Die in Kapitel 2.1.1 beschriebenen Pain-Scales geben wenig Informationen über „typische Beobachtungszeiträume von Pain-Scales“, da sie in den meisten Fällen in den Anleitungen nicht beschrieben werden. Bei der FLACC-Scale wird empfohlen, das Baby eine bis fünf Minuten zu beobachten.[45] Es gibt keine belastbare Grundlagen, um Werte für t_{obs} vorzuschalgen. Wie bei der Festlegung des Aktualisierungsintervalls ist es möglich, den Wert t_{obs} von den medizinischen Fachkräften selbstständig festlegen zu lassen, oder fest an die verwendete Pain-Scale zu binden. Eine weitere Variante ist, t_{obs} an den Wert des Parameters zu binden t_{act}, damit das medizinische Personal nur einen Wert festlegen muss. Ein Verhältnis von $t_{obs} = k \cdot t_{act}$ würde mit $k = 1$ nicht-überlappende Beobachtungszeiträume und mit $k = 2$ überlappende Beobachtungszeiträume erzeugen.

6 Zusammenfassung

Literaturverzeichnis

[1] Tobias Kaufmann Beat Pfister. *Sprachverarbeitung*. Springer, Berlin, 2008.

[2] Arthur H Benade. Fundamentals of Musical Acoustics. 1976.

[3] Judy Bildner. *CRIES Instrument Assessment Tool of Pain in Neonates*. City of Hope Pain, 1997. Online unter `http://prc.coh.org/pdf/CRIES.pdf`.

[4] Richard Brown. The short time fourier transform, 2014. Online erhältlich unter: `http://spinlab.wpi.edu/courses/ece503_2014/12-6stft.pdf`.

[5] R Sisto & Giuseppe Buonocore Carlo Bellieni, Franco Bagnoli. Cry features reflect pain intensity in term newborns: An alarm threshold. *Pediatric Research*, 5:142–146, 1. Online unter `https://www.researchgate.net/publication/297827342_Cry_features_reflect_pain_intensity_in_term_newborns_An_alarm_threshold`.

[6] Rami Cohen and Yizhar Lavner. Infant Cry Analysis and Detection. In *27th Convention of Electrical and Electronics Engineers in Israel*. IEEE, 2012. Online unter `https://www.researchgate.net/publication/261116332_Infant_cry_analysis_and_detection`.

[7] Alin Dobra. Introduction to classification and regression, 2005. Online erhältlich unter: `https://www.cise.ufl.edu/~adobra/datamining/classif-intro.pdf`.

[8] H. Hollien & T Murry E Müller. Perceptual responses to infant crying: identification of cry types. *Journal of Child Language*, 1(1):89–95, 1974. Online unter `https://www.cambridge.org/core/journals/journal-of-child-language/article/perceptual-responses-to-infant-crying-identification-of-cry-types/4F0F8088116FCE381851D8D560697A5F`.

[9] Jan Hamers Eva Cignac, Romano Mueller and Peter Gessler. Pain assessment in the neonate using the Bernese Pain Scale for Neonates. *Early Human Development*, 78(2):125–131, 2004. Online unter `https://www.researchgate.net/publication/8485535_Pain_assessment_in_the_Neonate_using_the_Bernese_Pain_Scale_for_Neonates`.

[10] Barbara Fuller. Acoustic Discrimination of three Cry Types. *Nursing Research*, 40(3), 1991. Online erhältlich unter: `https://www.researchgate.net/publication/21125005_Acoustic_Discrimination_of_Three_Types_of_Infant_Cries`.

[11] Trevor Hastie Gareth James, Daniela Witten and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.

[12] Dmitry Goldgof Rangachar Kasturi Terri Ashmeade Ghada Zamzmi, Chih-Yun Pai and Yu Sun. An Approach for Automated Multimodal Analysis of Infants' Pain. In *23rd International Conference on Pattern Recognition*, Cancun, Mexico, 2016.

[13] Dmitry Goldgof Rangachar Kasturi Yu Sun Ghada Zamzmi, Chih-Yun Pai and Terri Ashmeade. Machine-based Multimodal Pain Assessment Tool for Infants: A Review,

2016. Online unter https://arxiv.org/ftp/arxiv/papers/1607/1607.00331.pdf.

[14] Ricardo Gutierrez-Osuna. Introduction to Speech Processing. Online unter http://courses.cs.tamu.edu/rgutier/csce689_s11/.

[15] Health Facts For You. *Using Pediatric Pain Scales Neonatal Infant Pain Scale (NIPS)*, 2014. Online unter https://www.uwhealth.org/healthfacts/parenting/7711.pdf und unter https://com-jax-emergency-pami.sites.medinfo.ufl.edu/files/2015/02/Neonatal-Infant-Pain-Scale-NIPS-pain-scale.pdf.

[16] Hodgkinson. Neonatal Pain Assessment Tool, 2012. Online unter http://www.rch.org.au/rchcpg/hospital_clinical_guideline_index/Neonatal_Pain_Assessment/#The%20Pain%20Assessment%20Tool.

[17] Michael J Corwin Howard L Golub. A Physioacoustic Model of the Infant Cry. In *Infant Crying - Theoretical and Research Perspectives*, chapter 3, pages 59 – 82. Plenung, 1985.

[18] Bonnie Stevens Huda Huijer Abu-Saad, Gerrie Bours and Jan Hamers. Assessment of pain in Neonates. *Seminars in Perinatology*, 2(5):402–416, 1998. Online unter https://www.ncbi.nlm.nih.gov/pubmed/9820565.

[19] Donna Geiss Laura Wozniak & Charles Hall Ivan Hand, Lawrence Noble. COVERS Neonatal Pain Scale: Development and Validation. *International Journal of Pediatrics*, 2010, 2010. Online unter https://www.hindawi.com/journals/ijpedi/2010/496719/.

[20] J Gorriz & J Segura J Ramorez. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness. *Robust Speech Recongnition and Understanding*, page 460, 2007. Online unter http://cdn.intechopen.com/pdfs/104/InTech-Voice_activity_detection_fundamentals_and_speech_recognition_system_robustness.pdf.

[21] Jeih-weih Hung & Lin-shan Lee Jia-lin Shen. Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments. 1998. Online unter https://www.researchgate.net/publication/221489354_Robust_entropy-based_endpoint_detection_for_speech_recognition_in_noisy_environments.

[22] Carol Espy-Wilson & Tarun Pruthi Jonathan Kola. Voice Activity Detection. *MERIT BIEN*, 2011. Online unter http://www.ece.umd.edu/merit/archives/merit2011/merit_fair11_reports/report_Kola.pdf.

[23] Bonnie J. Stevens K. J. S. Anand and Patrick J. McGrath. *Pain in Neonates and Infants*. Elsevier, 2007.

[24] Kim Weaver & Fathi M. Salam Khurram Waheed. A robust Algorithm for detecting speech segments using an entropic contrast. *IEEE*, 2003. Online unter http://ieeexplore.ieee.org/document/1187039/.

[25] Miroslav Kubat. *An Introduction to Machine Learning*. Springer, 2015.

[26] Barry Lester and Zachariah Boukydis. *Infant Crying: Theoretical and Research Perspectives*. Springer, 1985.

[27] A. Rebecca Neal Linda L. LaGasse and Barry M. Lester. Assessment of infant cry: Acoustic cry analysis and parental perception. *Mental retardation and developmental disabilities*, 11(1):83–93, 2005. Online unter https://www.ncbi.nlm.nih.gov/pubmed/15856439.

[28] Tze-Wey Loong. Understanding sensitivity and specificity with the right side of the brain. *BMJ*, 327(7417), 2003. Online unter `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC200804/`.

[29] Michael Lutter. Speech production, 2015. Online erhältlich unter: `http://recognize-speech.com/speech/speech-production`.

[30] M M Homayounpour M H Moattar. A simple but efficient real-time Voice Activity Detection Algorithm. Signal Processing Conference, IEEE, August 2009. Online unter `http://ieeexplore.ieee.org/document/7077834/?arnumber=7077834&tag=1`.

[31] Robert Mannell. Acoustic theory of speech production, 2015. Online erhältlich unter: `http://clas.mq.edu.au/speech/acoustics/frequency/acoustic_theory.html`.

[32] Stephen Marsland. *Machine Learning - An Algorithmic Perspective*. Chapman & Hall / CRC, 2009.

[33] Tom M Mitchell. *Machine Learning*. WCB McGraw-Hill, 1997.

[34] Hans M Koot Dick Tibboel Jan Passchier & Hugo Duivenvoorden Monique van Dijk, Josien de Boer. The reliability and validity of the COMFORT scale as a postoperative pain instrument in 0 to 3-year-old infants. *Pain*, 84(2):367—-377, 2000. Online unter `http://www.sciencedirect.com/science/article/pii/S0304395999002390` und unter.

[35] Sinno Simons Monique van Dijk and Dick Tibboel. Pain assessment in neonates. *Paediatric and Perinatal Drug Therapy*, 6(2):97–103, 2004. Online unter `http://www.sciencedirect.com/science/article/pii/S0304395999002390`.

[36] D L Neuhoff. *Signal and Systems I - EECS 206 Laboratory*. The University of Michigan, 2002. Online erhältlich unter: `http://www.eecs.umich.edu/courses/eecs206/archive/spring02/` abgerufen am 11. Januar 2016.

[37] Katarina Michelsson Ole Wasz-Hockert and John Lind. Twenty-Five Years of Scandinavian Cry Research. In *Infant Crying - Theoretical and Research Perspectives*, chapter 3, pages 59 – 82. Plenung, 1985.

[38] J L Mathew P J Mathew. Assessment and management of pain in infants. *Postgrad Med J*, 79:438–443, 2003. Online unter `http://pmj.bmj.com/content/79/934/438.full`.

[39] Steven Creech Patricia Hummel, Mary Puchalski and Marc Weiss. N-PASS: Neonatal Pain, Agitation and Sedation Scale – Reliability and Validity. *Pediatrics/Neonatology*, 2(6), 2004. Online unter `http://www.anestesiarianimazione.com/2004/06c.asp`.

[40] Susan Parker-Price & Ronald Barr Philip Zeskind. Rythmic organization of the Sound of Infant Cry. *Dev Psychobiol*, 26(6):321–333, 1993. Online unter `https://www.ncbi.nlm.nih.gov/pubmed/8119482`.

[41] Ananth N. Iyer Pritam Pal and Robert E. Yantorno. Emotion detection from infant facial experessions and cries. In *Acoustics, Speech and Signal Processing*. IEEE, 2006.

[42] R Ward & C Laszlo Qiaobing Xie. Automatic Assessment of Infants Levels-of-Distress from the Cry Signals. *IEEE Transanctions on Speech and Audio Processing*, 4(4):253–265, 1996. Online unter `http://ieeexplore.ieee.org/document/506929/`.

[43] Brian Hopkins & James Green Ronald Barr. *Crying as a Sign, a Symptom, and a Signal*. Mac Keith Press, 2000.

[44] Juan Ignacio Godino-Llorente Ruben Fraile. Cepstral peak prominence: A comprehensive analysis. *Biomedical Signal Processing and Control*, 14(1):42–54, 2014. Online unter https://www.researchgate.net/publication/264084923_Cepstral_peak_prominence_A_comprehensive_analysis.

[45] J R Shayevitz & Shobha Malviya Sandra Merkel, Terri Voepel-Lewis. The FLACC: A Behavioral Scale for Scoring Postoperative Pain in Young Children. *Pediatric Nursing*, 23(3):293–7, 1996. Online unter https://www.researchgate.net/publication/13998379_The_FLACC_A_Behavioral_Scale_for_Scoring_Postoperative_Pain_in_Young_Children.

[46] Andreas Spanias Sassan Ahmadi. Cepstrum-Based Pitch Detection Using a New Statistical V/UV Classification Algorithm. *IEEE Transactions on Speech and Audio Detection*, 7(3):333–338, 1999. Online unter http://ieeexplore.ieee.org/document/759042/.

[47] Julius Smith. *Spectral Audio Signal Processing*. Center for Computer Research in Music and Acoustics (CCRMA), 1993. Online unter https://www.dsprelated.com/freebooks/sasp/Short_Time_Fourier_Transform.html.

[48] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1999. Online erhältlich unter: http://www.dspguide.com/pdfbook.htm.

[49] Henning Reetz & Carla Wegener Tanja Fuhr. Comparison of Supervised-learning Models for Infant Cry Classification. *InternatIonAl Journal of HealtH Professions*, 2015. Online unter https://www.degruyter.com/view/j/ijhp.2015.2.issue-1/ijhp-2015-0005/ijhp-2015-0005.xml.

[50] Sabine Deligne & Peder Olsen Trausti Kristjansson. Voicing Features for Robust Speech Detection. In *Interspeech Lisboa*, September 2005. Online unter http://papers.traustikristjansson.info/wp-content/uploads/2011/07/KristjanssonRobustVoicingEurospeech2005.pdf.

[51] Gyorgy Ivan Varallyay. *Analysis of the Infant Cry with Objective Methods*. PhD thesis, Budapest University of Technology and Economics, 2009. Online erhältlich unter: https://pdfs.semanticscholar.org/5c38/b368dc71d67cbfabc3077a50536b086d8eec.pdf.

[52] P H Wolff. The role fo biological rhythms in early psychological development. *Bulletin of the Menninger Clinic*, 31(1):197–218, 1967.

[53] Syed Ahmad Yousra Abdulaziz, Sharrifah Mumtazah. Infant Cry Recognition System: A Comparison of System Performance based on Mel Frequency and Linear Prediction Coefficients. In *Information Retrieval & Knowledge Management*, 2010. Online unter http://ieeexplore.ieee.org/document/5466907/.

Appendices

Tabelle .1: Accuracy-Werte der Grenzwertfindung mit REPTree

$SNR_{Training}$	3 dB				50 dB				50+3 dB			
SNR_{Test}	3 dB	50 dB	7 dB*	Mean	3 dB	50 dB	7 dB*	Mean	3 dB	50 dB	7 dB*	Mean
Zeit	77.81%	79.02%	86.04%	80,96%	49.33%	94.70%	48.66%	64,23%	77.54%	92.47%	84.38%	84,80%
Freq	82.05%	89.28%	82.71%	84,68%	70.52%	94.37%	55.06%	73,31%	81.75%	91.22%	74.90%	82,62%
Ceps	88.98%	94.72%	92.96%	**92,22%**	86.83%	94.68%	92.83%	**91,45%**	88.98%	94.72%	92.96%	**92,22%**
Corr	80.45%	73.47%	84.89%	79,60%	73.07%	87.14%	77.98%	79,39%	77.90%	84.88%	82.84%	81,87%
Zeit+Freq	82.05%	89.28%	82.71%	84,68%	70.52%	94.37%	55.06%	73,31%	81.75%	91.22%	74.90%	82,62%
Zeit+Ceps	88.98%	94.72%	92.96%	**92,22%**	86.83%	94.68%	92.83%	**91,45%**	88.98%	94.72%	92.96%	**92,22%**
Zeit+Corr	80.45%	73.47%	84.89%	79,60%	49.33%	94.70%	48.66%	64,23%	80.32%	92.35%	88.22%	86,96%
Freq+Ceps	88.98%	94.72%	92.96%	**92,22%**	70.65%	94.75%	55.06%	73,49%	88.98%	94.72%	92.96%	**92,22%**
Freq+Corr	82.05%	89.28%	82.71%	84,68%	70.52%	95.60%	95.60%	87,24%	81.75%	94.42%	74.90%	83,69%

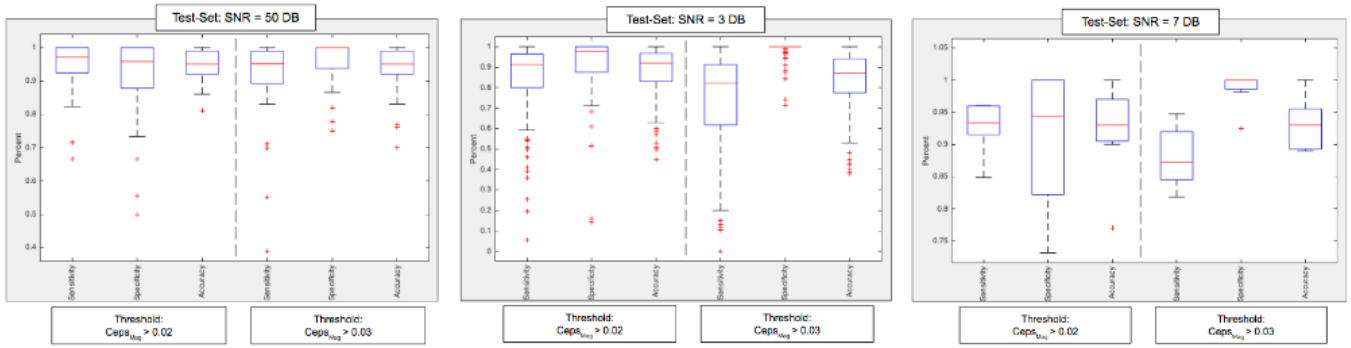

Abbildung .1: Boxplot-Auswertung über Sensitivity, Specificity und Accuracy der beiden VAD-Modelle