# 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 Neugeberonenen 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 die Schlüsse, die man aus dem Weinen eines Babys auf dessen Schmerz machen kann. Um die menschliche Stimme automatisiert analysieren zu können, werden Methoden der Signalverarbeitung verwendet. Daher werden in Kapitel 2.2 technische Grundlagen erläutert, die zu diesem Zweck unerlässlich sind. In Kapitel 2.3 wird eine Einführung in die „klassische Schreiforschung". Dabei handelt es sich um Wissenschaftsgebiet, bei dem versucht wird, mit Hilfe von Methoden der Signalverarbeitung ein tieferes Verständnis über die Bedeutung des Weinens von Babys zu erhalten. Da sich in dieser Arbeit erstellte Konzept zur automatisierten Analyse des Weinenes als Erweiterung der klassischen Methoden versteht, ist ein Verständnis des Wissenschaftsgebietes unerlässlich. In Kapitel 2.4 werden Grundlagen des Überwachten maschienellen Lernens erläutert, da diese zur automatisierten Interpretation von Audiosignalen von Bedeutung sind.

## 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".[39, S. 438] Abseits von dieser theoretischen Definition hat der Mensch ein intuitives Verständnis für Schmerz, da jeder ihn in seine Leben erfahren musste. In der ersten Hälfte des 20sten Jahrhunderts war die vorherrschende Meinung, dass Neugeborene keinen Schmerz empfinden können. Beispielsweise bekam sie nach Operationen keine Schmerzmittel verabreicht. Die aktuell vorherrschende Meinung ist, 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] [39, S. 438]

### 2.1.1 Pain Scales

Es gibt diverse Gründe, warum Neugeborene Schmerz empfinden 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.[39, S. 441]

Schlussendlich ist Schmerz jedoch ein subjektives Empfinden. 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.[39, S. 438] Die Schmerzdiagnostik durch eine andere Person ist etwas inherent subjektives und abhängig von Faktoren wie dem Alter, Geschlecht, kulturellen Hintergrund, persönlichen Erfahrungen mit Schmerz etc.[13, S. 3] Um die Schmerzdiagnostik objektiver zu gestalten, wurden daher sogenannte *Pain Scales* entwickelt, mit Hilfe eines Punktesystems den Schmerzgrad des Babys quantifizieren.[39, S. 438 - 439] Es existieren *monomodale* oder *unidimensionale* Pain Scales, bei denen der Schmerzgrad aus der Beobachtung *eines* Merkmals geschlossen wird, wie beispielsweise der Gesichtsausdruck. Ein Merkmal wird in diesem Zusammenhang als *Schmerzindikator* bezeichnet. *Multimodale* oder auch *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. Für jede aufgeführte Kategorie 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]

Tabelle 2.1: Neonatal Infant Pain Scale [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 oder dem konkreten Einsatzzweck. Einige Pain Scales sind beispielsweise auf die Schmerzdiagnostik während eines Eingriffes spezialisiert, andere auf den darauf folgenden Heilungsprozess. In den meisten 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. In der Übersicht wird nur der Teil wiedergegeben, der sich auf das Weinen bezieht. Es wird nicht gezeigt, welche weiteren Merkmale in das jeweilige Scoring mit einbezogen werden, für welchen Altersbereich die Scales gedacht sind oder welches Scoring auf welche Schmerzintensität hinweist. Es soll an dieser Stelle nur verdeutlicht werden, welche Ansätze zur Bewertung des Weinens aus medizinischer Sicht im Zusammenhang mit Pain Scales existieren.

| System | P. | Description |
|---|---|---|
| FLACC***[45] | 0 | No cry (awake or asleep) |
| | 1 | Moans or whimpers; occasional complaint |
| | 2 | Crying steadily, screams or sobs, frequent complaints |
| N-PASS***[40] | -2 | No cry with painful stimul |
| | -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 |
| BIIP[9] | 0 | No Crying |
| | 1 | Crying <2 minutes |
| | 2 | Crying >2 minutes |
| | 3 | Shrill Crying >2 minutes |
| CRIES*[3] | 0 | If no cry or cry which is not high pitched |
| | 1 | If cry high pitched but baby is easily consoled |
| | 2 | If cry is high pitched and baby is inconsolable |
| COVERS**[19] | 0 | No Cry |
| | 1 | High-Pitched or visibly crying |
| | 2 | Inconsolable or difficult to soothe |
| PAT*[16] | 0 | No Cry |
| | 1 | Cry |
| DAN**[5] | 0 | Moans Briefly |
| | 1 | Intermittent Crying |
| | 2 | Long-Lasting Crying, Continuous howl |
| COMFORT*[34] | 0 | No crying |
| | 1 | Sobbing or gasping |
| | 2 | Moaning |
| | 3 | Crying |
| | 4 | Screaming |
| MBPS[37] | 0 | Laughing or giggling |
| | 1 | Not Crying |
| | 2 | Moaning quiet vocalizing gentle or whimpering cry |
| | 3 | Full lunged cry or sobbing |
| | 4 | Full lunged cry more than baseline cry |

Tabelle 2.2: Übersicht über Pain-Scales. Legende zu den Einsatzbereichen: *** Anhaltender/chronischer Schmerz, ** Prozeduraler Schmerz, *Post-Operativer Schmerz[35, S. 98 ]

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.

Folgende Anmerkungen werden bezüglich der Pain Scales aus Tabelle 2.2 gemacht:

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 in die Anwendungsvorschriften der Pain Scales werden keine festen Definitionen gegeben. Dies erleichtert den praktischen Einsatz der Pain Scales, führt jedoch zu einem Interpretationsspielraum und somit zu einem von der diagnostizierenden Person abhängigen Scoring. Die *BIIP*-Scale nutzt als einzige der vorgestellten Scales objektiv messbare Eigenschaften.
2. Die Pain Scales fokussieren unterschiedliche Eigenschaften. Bei *CRIES* ist die Tonhöhe, bei *BIIP* die Länge und bei *COMFORT* die Art des Weinens ausschlagebend für das 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 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. Diese Unterschiede seien im Spektogramm sichtbar. Wenige Jahre Später zeigten Müller et al. [8], dass bei leichter Veränderung des Experimentes die Unterscheidung nicht mehr möglich sei. Die Gegenhypothese ist, dass Weinen „nichts als undifferenziertes Rauschen“ sei. 50 Jahre später liegt kein anerkannter Beweis für die eine oder andere Hypothese vor. Es gibt lediglich starke Hinweise dafür, dass die Plötzlichkeit des Eintretens des Grundes sich 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.[44, 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 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 Kindes 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. [44, S. 13 - 17] [43] 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-Scala (siehe Tabelle 2.2) die Tonhöhe steigt. Qiaobing Xie et al. [43] haben festgestellt, dass häufiges und „verzerrtes“ Schreien (ohne feststellbares Grundfrequenz, da der Ton stimmlos erzeugt wird) 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 beschrieben werden kann. Möchte man das Weinen objektiv beschreiben und messbar machten, so verwendet man die Methoden der digitalen Signalverarbeitung. An dieser Stelle wird eine Einführung in die wichtigsten Themen dieses Wissenschaftsgebietes gegeben, die im Zusammenhang mit der Audiosignalverarbeitung größere Bedeutung haben. Es wird ein grundlegendes Verständnis der Signalverarbeitung vorausgesetzt, da die Erläuterungen in diesem Kapitel eher der Definition der verwendeten Begriffe dient, aus Platzgründen jedoch keine für Neulinge geeignete Einführung in das Themengebiet gewährleisten 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 diskreten 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 (in diesem Zusammenhang auch als *Sample* bezeichnet) an dem Index $\hat{=}$ Zeitpunkt $n$. Die Samplingfrequenz des digitalen Signales 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 (ein „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 Signales 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 die häufig verwendete Signaleigenschaften gegeben. Abbildung 2.1 visualisiert die Erläuterungen.

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

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

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 an dieser Stelle 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]

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

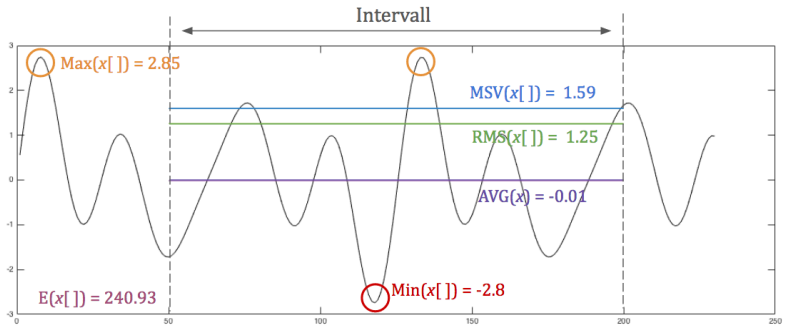


Abbildung 2.1: Statistische Werte eines Signals über das Intervall [50,200]

### 2.2.3 Fehlersignale

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

$$x'[\,] := \underset{n=n_1}{\overset{n_2}{\forall}} : x'[n] = x[n] + e[n] \tag{2.9}$$

Kennt man sowohl das Eingangssignal $x[\,]$ als auch das Ausgangssignal $x'[\,]$, kann das Störsignal $e[\,]$ nach Formel 2.10 berechnet werden.

$$e[\,] := \underset{n=n_1}{\overset{n_2}{\forall}} : e[n] = x'[n] - x[n] \tag{2.10}$$

Eine Möglichkeit der Quantifizierung der Stärke des Rauschens auf das Signal ist, das Eingangssignal ins Verhältnis zum Rauschsignal zu setzen. Formel 2.11 gibt die Definition.

$$\mathrm{SNR}_{rel}(x[\,], e[\,]) = \frac{MSV(x[\,])}{MSV(e[\,])} \tag{2.11}$$

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

2.12 definiert den *Signal/Rausch-Abstand* (*SNR*, englisch Signal-to-Noise-Ratio). Ein *niedriger* SNR-Wert auf ein *starkes* Rauschen hin, und ein *hoher* SNR auf ein *schwaches* Rauschen. Abbildung **??** visualisiert die Berechnung des SNR. Im Zusammenhang mit der Spracherkennung ist der Signal/Rausch-Abstand von Bedeutung, da ein höheres Rauschen die Verarbeitung des Nutzssignals, der Sprache, erschwert.

$$\mathrm{SNR}(x[\,], e[\,]) = 10 \cdot \lg\left(\frac{MSV(x[\,])}{MSV(e[\,])}\right) \text{ dB} \tag{2.12}$$

### 2.2.4 Kurzzeit-Fourier-Transformation

Das Signal $x[\,]$ beschreibt den Zeitbereich des Signals, da die unabhängige Variable die Zeit definiert. Gleichung 2.13 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]

$$\mathrm{DFT}\{x[\,]\} = X[\,] := \quad \underset{k=0}{\overset{N-1}{\forall}} : \; X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi k \frac{n}{N}} \tag{2.13}$$

Das *Spektrum* wird nach Gleichung 2.14 definiert als der Absolutwert des Frequenzbereiches im Bereich $0, \ldots, N/2$.

$$\text{Spektrum} := \quad |X[0]|, \; \ldots \; , |X[N/2]| \tag{2.14}$$

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 klar erkennen, zu welchen Zeitpunkten die Töne beginnen und Enden, aber nicht, welche Frequenzen in den Tönen enthalten sind. Unten ist der 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 logarithmisiert dargestellt. Das Frequenzspektrum zeigt, welche Frequenzen im dem Signal enthalten sind. So sind beispielsweise keine Frequenz unterhalb von 1000 Hz enthalten. Das Spektrum acht jedoch nicht erkennbar, welche Frequenzen enthalten sind.

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 $x_i[\,]$. Dieser Prozess wird als *Windowing* bezeichnet.[47]

$$x_i[\,] := \quad \underset{n=0}{\overset{M-1}{\forall}} : \; x_m[n] = x[n + i \cdot R] \tag{2.15}$$

Abbildung 2.3 gibt ein Beispiel für die Zerlegung von $x$ in Signalfenster $x_0[\,], \ldots, x_4[\,]$.

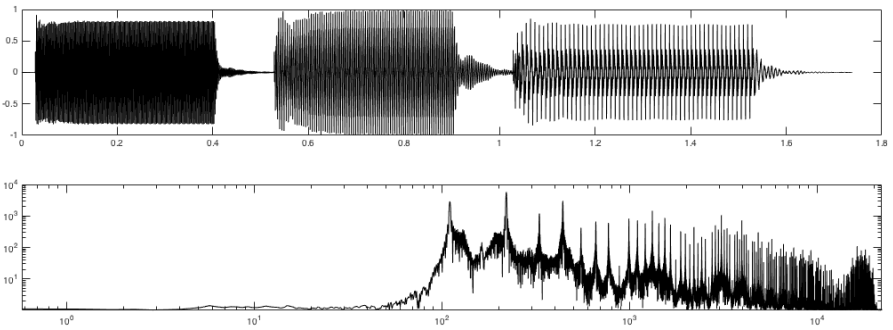

Abbildung 2.2: Ein 1.8-Sekunden langes Signal. Oben: Der Zeitbereich mit drei klar erkennbaren Events. Unten: Das Frequenz-Spectrum des gesamten Signals mit logarithmisierten Achsen.

Die Samplingrate des Signals ist $f_s = 44100$, die Fensterlänge beträgt $M = 22050/f_s = 0.5\,\text{s}$ und die Hoptime $R = M/2 = 0.25\,\text{s}$.

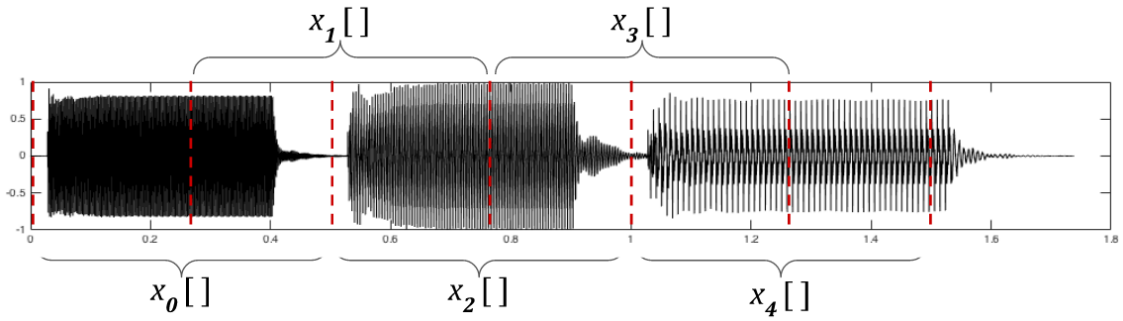


Abbildung 2.3: Windowing: Die Zerlegung eines Signals in kürzere Fenster.

Als Vorbereitungsschritt für die Transformation der Signalfenster in den Frequenzbereich wird nun jedes Fenster mit einer sogenannten *Fensterfunktion* (engl *window*) $w[\,]$ multipliziert.[1, S. 69] Gleichung 2.16 definiert eine der am weitesten verbreiteten Fenster-Funktionen, das *Hamming-Window*. Der Paramter $M$ gibt die länge des Fensters an. Abbildung 2.4 visualisiert das Hamming-Window. [48, S. 286]

$$w[\,] := \quad \forall_{n=0}^{M-1} : \; w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{M}) \tag{2.16}$$

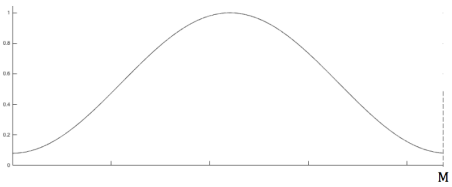


Abbildung 2.4: Das Hamming-Window

Die Gleichung 2.17 definiert die *Kurzzeit-Fourier-Transformation* (engl *Short Time Fourier Transformation*, *STFT*), implementiert mit Hilfe der DFT. Dabei wird das 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.

$$\mathrm{STFT}_i\{x[\,]\} = X_i[\,] := \bigvee_{k=0}^{M-1} : \; X_i[k] = \sum_{n=0}^{M-1} x[n + i \cdot R] \cdot w[n] \cdot e^{-j2\pi k \frac{n}{N}} \tag{2.17}$$

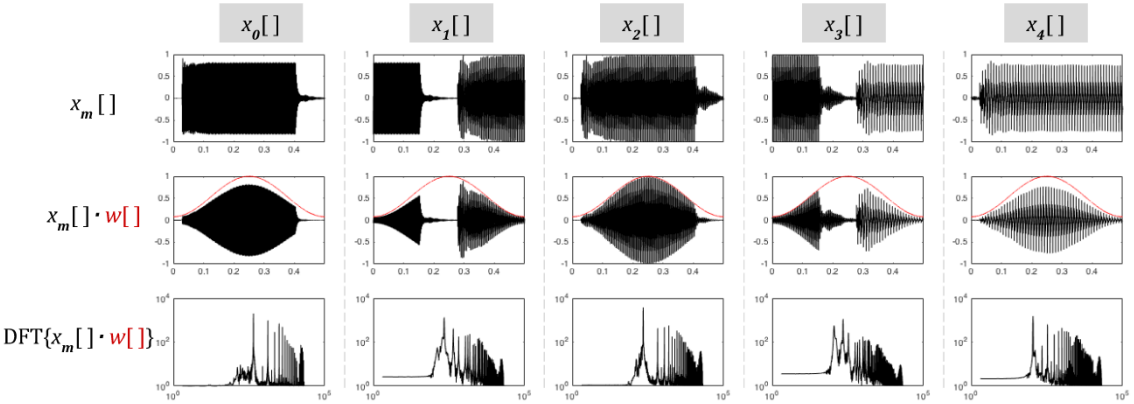


Abbildung 2.5: STFT des Beispiel-Signals aus Abbildung 2.3

### 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 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 äußert. 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“ beschrieben werden. Das Halszäpfchen bestimmt, ob der Luftstrom in den Mund- oder Nasenraum geleitet wird. Die Stellung der Artikulatoren, bestehend aus 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.

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 Eingangs-Signal $u[\,]$ bezeichnet. Dieses Signal wird daraufhin an den Vokaltrakt weitergeben, welcher als

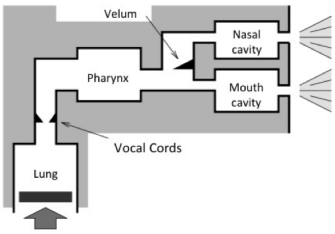


Abbildung 2.6: Schematische Übersicht über die Organe der Spracherzeugung. Lung = Lunge, Vocal Chords = Stimmbänder, Pharynx = Rachen, Velum = Halszäpfchen, Mouth Cavity = Mundraum, Nasal Cavity = Nasenraum [29]

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 als die Faltung des Signals $u[\,]$ und den beiden linearen, zeitinvarianten Filtern nach Gleichung 2.18. Gleichung 2.19 definiert den Frequenzbereich des Ausgangssignals $Y[\,]$ durch die Multiplikation der Frequenzbereiche dieser drei Komponenten. Abbildung 2.7 visualisiert diesen Prozess schematisch. [17, S. 62 - 63] [29]

$$u[\,] * v[\,] * r[\,] = y[\,] \tag{2.18}$$

$$U[\,] \cdot V[\,] \cdot R[\,] = Y[\,] \tag{2.19}$$

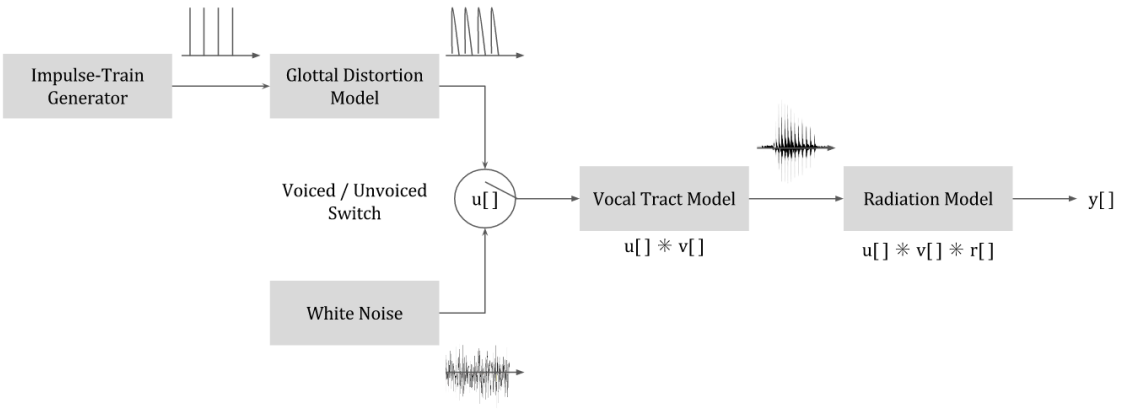


Abbildung 2.7: Schematische über das Source-Filter-Model [14, nach Source estimation, S. 17]

Abbildung 2.8 zeigt die Zeitbereiche des stimmhaften und turbulenten Signals 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 2.9 zeigt die Frequenzbereiche der Komponenten des Source-Filter-Modells. Die periodische Quelle ($U[\,]$ links) zeichnet sich im Frequenzbereich durch gleichmäßig verteilte

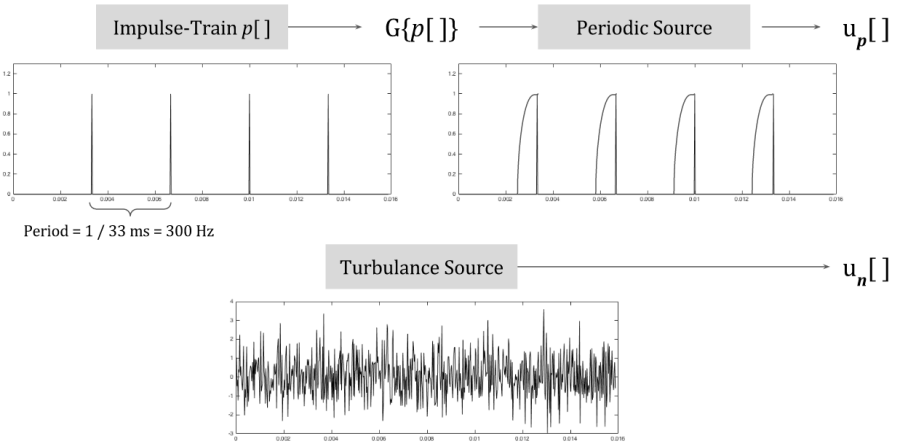


Abbildung 2.8: Zeit-Bereiche der periodic und der turbulance Source [31, Source]

Spitzen aus, die mit steigender Frequenz an Amplitude verlieren. Rechts daneben ist der Frequenzbereich des weißen Rauschen zu sehen. 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 2.10 zeigt schematisch das Spektrum eines stimmhaften Sprachsignals. Sowohl die Grundfrequenz als auch die harmonischen Obertonwellen sind rein visuell als „vielen, kurzen Signalspitzen“ im Spektrum erkennbar. Der kleinste gemeinsame Teiler der Frequenzen dieser Signalspitzen entspricht der Grundfrequenz $f_0$ dieses Stimmsignals, in diesem Beispiel 250.7 Hz. Die Grundfrequenz ist ebenfalls an der Signalspitze mit der tiefsten Frequenz ablesbar. Die harmonsichen 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 geprüft wird (vergleiche mit Abbildung 2.9).[1, S. 52 - 53]

Abbildung 2.11 verdeutlicht, wie der als lineares, zeitinvariantes Filter modellierte Vokaltrakt durch Formanten beschrieben 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 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 stärkster zur Identifikation und Unterscheidung der Vokale genutzt werden. 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]

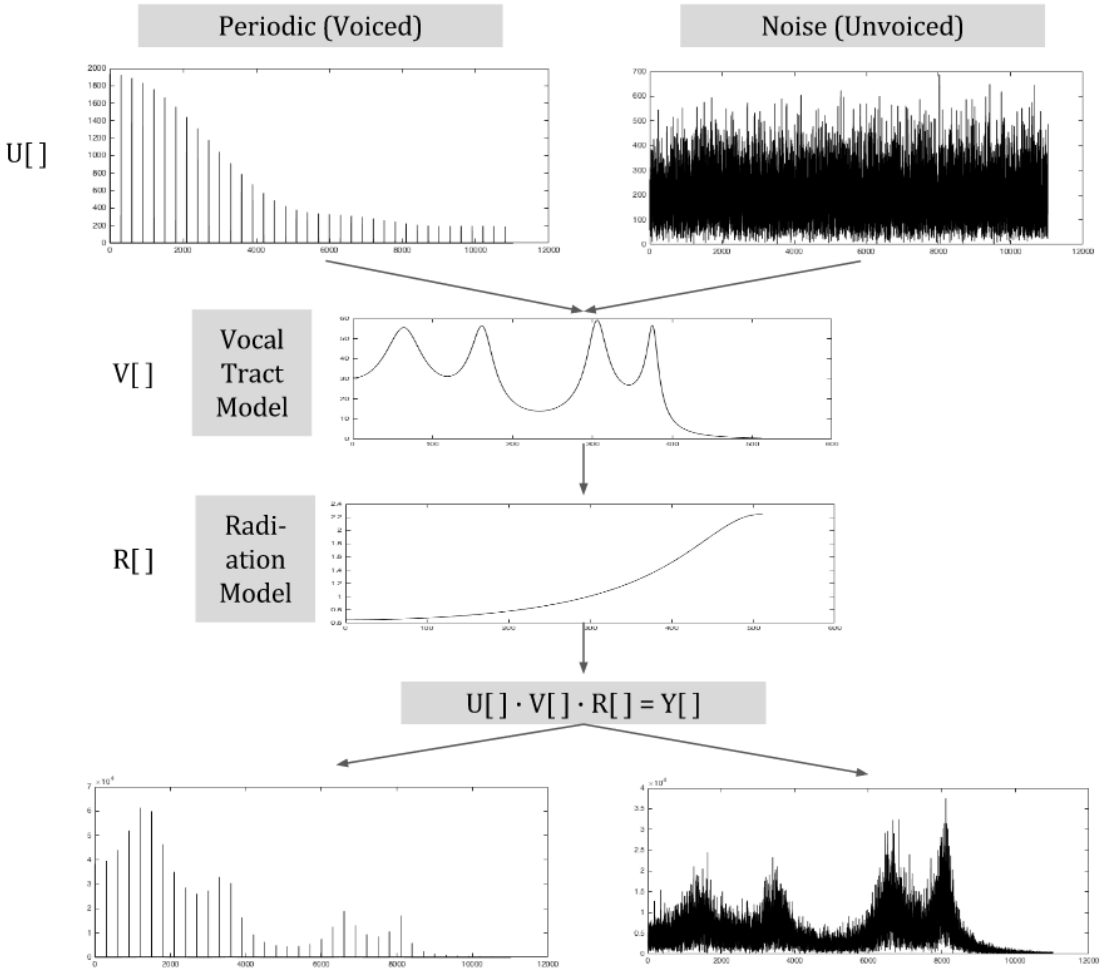


Abbildung 2.9: Betrachtung der Frequenz-Bereiche des Source-Filer-Modell (nach: [14, Source Estimation, S. 3])

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 über kurze Zeitbereiche weniger Millisekunden als periodisch. Schlussendlich ist die Stimme nie perfekt periodisch, 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 Spektograms. 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.

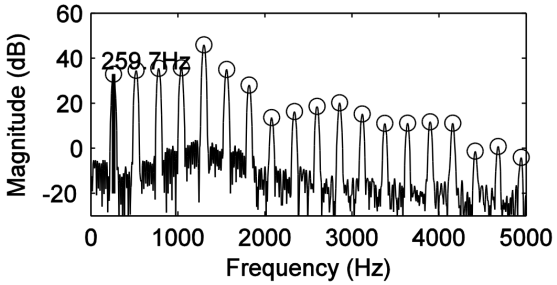


Abbildung 2.10: Grundfrequenz und harmonische Obertöne eines Sprachsignals.

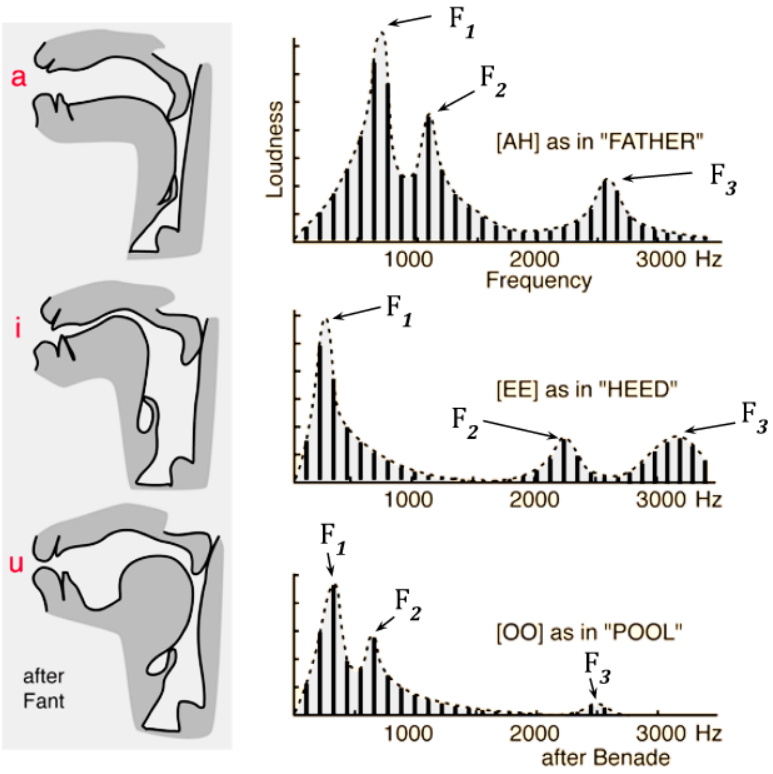


Abbildung 2.11: Formanten im Sprach-Signal (nach: [2])

Es ist zu erkennen, wie bei der geringeren Fensterlänge der zeitliche Verlauf besser erkennbar, jedoch die einzelnen harmonsichen 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.

## 2.3 Schreiforschung

Das Wissenschaftsgebiet, welches sich mit der Analyse und Interpretation von Lautäußerungen Neugeborener auseinandersetzt, wird als „Schreiforschung“ bezeichnet. Das bis heute wohl prominenteste Forschungsgruppe dieses Wissenschaftsgebietes ist die im vergangenen Kapitel erwähnte „Scandinavian Cry-Group“[26], welche zwischen 1960 und 1990 die Laute von Babys systematisch erforscht haben. Das wichtigste Werkzeug zur Analyse der Lautäuße-

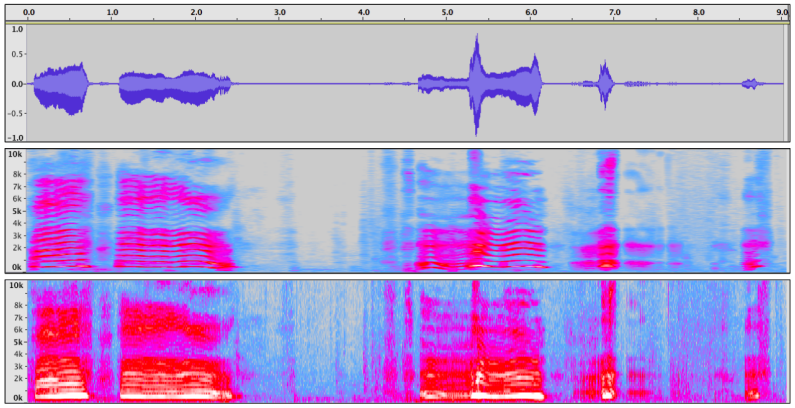

Abbildung 2.12: Spectogram von Baby-Weinen. Rot = Hohe Amplituden, Blau = niedrige Amplituden. Oben: Zeit-Bereich. Mitte: Spectogram mit einer Fensterlänge von 185 ms(8192-Sample DFT). Unten: Spectogram mit einer Fensterlänge von 5 ms
(265-Sample DFT).

rungen war das eben vorgestellte 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.[44, 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 eine Teilmenge dieses Vokabulars an dieser Stelle vorgestellt. Weiterhin werden Begriffe eingeführt, die von Zeskind et al. in „Rythmic organization of the Sound of Infant Cry “ veröffentlicht wurden.[41]

### 2.3.1 Phyisio-Akustische Modellierung des Weinens

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

Die folgenen 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. [41]. 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. Praktisch ergibt sich das Problem,

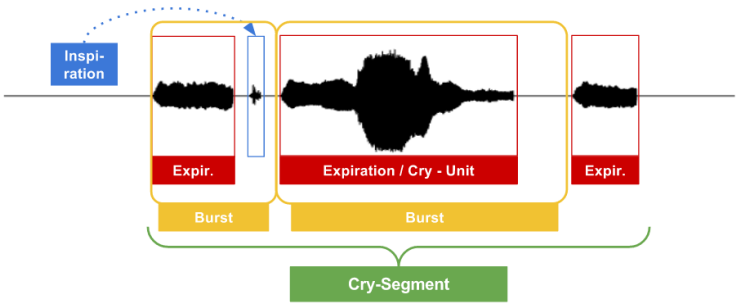


Abbildung 2.13: Veranschaulichung des Grundvokabulars

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:** 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.

Cry-Units werden von H Golub und M Corwin in eine der drei folgenen 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 umgangsprachlich 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 etc. 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.

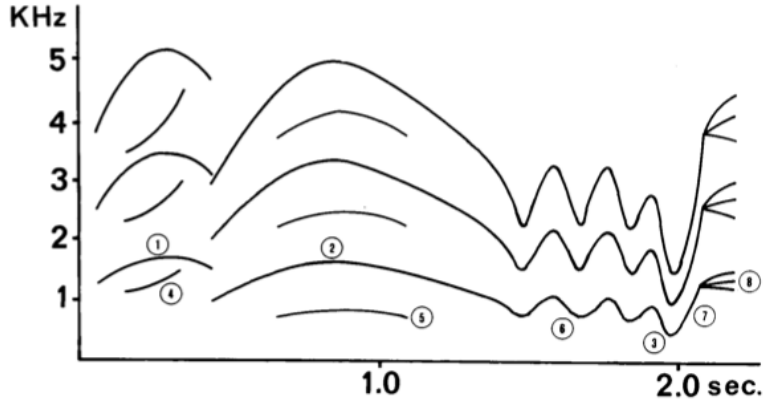


Abbildung 2.14: (1) Pitch of Shift (2) Maximale Grundfrequenz (3) Minimum der Grundfrequenz (4) Biphonation (5) Double Harmonic Break (6) Vibrato (7) Glide (8) Furcation [44, S. 142]

- **Amplitude:** Die Lautstärke der Cry-Unit, gemessen in Dezibel. [27, S. 85] [10, S. 156]

Golub et al. haben weiterhin eine Reihe von Features vorgestellt, die das zeitliche Verhalten der Grundfrequenz und der harmonischen Obertöne innheralb einer Cry-Unit beschreiben. [17, S. 73]

- **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 harmonsichen Obertöne in mehrere, schwächere Obertöne.

Abbildung 2.14 visualsiert diese Grundfrequenz bezogenen Features in einem schematisch dargstellten Spektogramm.

Die folgende Features 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 erster 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 durchschnittlichen Tonhöhen, Anzahl des Vorkommens bestimmter Melodiekonturen, Varianz der Länge der Cry-Units etc.[27, S. 85]

Verschiedene Krankheitsbilder wurden in Zusammenhang mit dem vorkommen bestimmter Features des Cry-Segmentes gebracht. So wurde eine Korrelation zwischen dem Anstieg der durchschnittlichen Grundfrequenz, häufiger Biphonation und geringer Duration in Zusammenhang mit Gehirnschäden gebracht. 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 [44, S. 142]:

- Es gibt keine Einigung darüber, welche der zahlreichen vorgestellten Eigenschaften die wichtigsten sind. Beispielsweise konzentrierten sich Golub et al. [17] vermehrt auf die Erkennung von Mustern im Melodieverlauf, Zeskind et al. auf zeitliche Eigenschaften. [41]. 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 vor allem von Golub et al. verwendet wurden [17], ist nicht einmal gesichert, ob es sich nicht doch um technische Artefakte der damals verwendeten Analogtechnik handelt. [27, S. 84 - 85]
- 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 verstarben, fast immer eine Erhöhung der Frequenz des ersten Formanten in Verbindung mit häufigen Cry-Mode-Changes zeigen. Viele Babys, die nicht am plötzlichen Kindstot versterben, zeigen jedoch die selben Merkmale.[27, S. 85]
- Selbst, wenn in verschiedenen Studien die selbe Eigenschaft verwendet wird, wie zum Beispiel die durchschnittliche Tonhöhe, ist nicht standardisiert, wie dieses exakt 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 etc.
- Golub et al. behaupten, bereits in den achziger 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 finden, welche für diese Arbeit von höchstem Interesse gewesen wären.

## 2.4 Klassifizierung und Regression

Klassifzierung und Regression sind Teilgebiete des Wissenschaftsgebietes des *Überwachten Lernens*, einem Teilgebiet des Wissenschaftsgebietes des *maschienellen Lernens*. Das Ziel der Überwachten Lernen ist es, ein *Prädikator* (*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.) das Gewicht und (2.) die Augenfarbe ist. Der Prädikator 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 werden. In Bezug auf das eben genannte Beispiel wäre das erste Feature $F_1 = \textit{Gewicht}$ und das zweite Feature $F_2 = \textit{Augenfarbe}$. Eine Instanz wäre in diesem Fall ein Tupel mit zwei beliebigen Werten dieser Attribute, wie zum Beispiel $x = (3\,\text{kg}, \text{Blau})$). 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 $D_{Traing}$ 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. $Y$ bezeichnet die Menge aller möglichen Outputs des Problems. Das heißt, $D_{Training} = \big( (x_1, t_1), \ldots (x_N, t_N) \big)$. 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 bisher bekannte oder unbekannten Trainings-Datensatzes $D_{Test}$ irrt. Der Test- und der Trainingsdatensatz 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.[1] [11, S. 28, 127]

[1] In vielen Quellen werden die Begriffe *Klassifizierung* und *Klassifikation* inkonsitent verwendet. Die *Klas-*

Bei der Regression *Regression* ist die Menge der möglichen Targets eines bestimmten Problem *kontinuierlich* und hat einen „quantitativen Charakter. Das heißt, es kann eine interne Ordnung in der Menge der Outputs festgelegt werden. Ein Beispiel für ein Regressionsproblem wäre die also die Ableitung des Alters des Babys, also $Y = \{0, \ldots, 130\}$. De 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 Klassfikators oder Prediktors. Welcher Algorithmus der „beste“ ist, das heißt für einen Test-Datensatz eine möglichst hohe *Genaugigkeit* 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*-Algorithmus, welcher genauer in Kapitel 2.4.1 beschrieben wird.

### 2.4.1 ID3

Es gibt drei Algorithmen zur Erzeugung von Entscheidungsbäumen, die weitreichende Einsatz finden: *ID3*, *C.45* und *CART*, wobei die letzteren Erweiterungen der grundlegenden Idee des *ID3*-Algorithmus darstellen. Daher wird an dieser Stelle zuerst der *ID3*-Algorithmus vorgestellt.

Es wird zunächst davon ausgegangen, dass alle Features diskret und nicht kontinuierlich sind. Tabelle 2.3 gibt einen Beispieldatensatz, an dessen Beispiel ein Classificator mit Hilfe des ID3 erzeugt wird. Es geht ähnlich dem Beispiel aus Tabelle **??** um die Frage, ob Federball-Spielen abhängig von Temperatur und Tageszeit Spaß macht, nur sind in diesem Fall alle Features diskret.

Tabelle 2.3: Beispieldatensatz D für die Kassfikation mit ID3

| $x_i$ | Temperatur | Tageszeit | $c_i$ = Spaß? |
|---|---|---|---|
| $x_1$ | warm | Tag | Ja |
| $x_2$ | kalt | Tag | Ja |
| $x_3$ | normal | Nacht | Nein |
| $x_4$ | kalt | Nacht | Nein |
| $x_5$ | normal | Tag | Ja |
| $x_6$ | warm | Nacht | Ja |

Abbildung 2.15 zeigt einen Klassifikator, den der ID-3 Algorithmus für diesen Datensatz baut. 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]

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 Feauture-Werte der betretenen 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*

*sifizierung* ist ein Prozess, dessen Ergebnis die *Klassifikation* ist. Daher wird von einem *Klassifizierungs-Algorithmus* gesprochen, da sich der Algorithmus auf den Prozess des Klassifizerens konzentriert, aber vom *Klassifikationsfehler*, da der Fehler des Ergebnisses der Klassifzierung bestimmt wird.

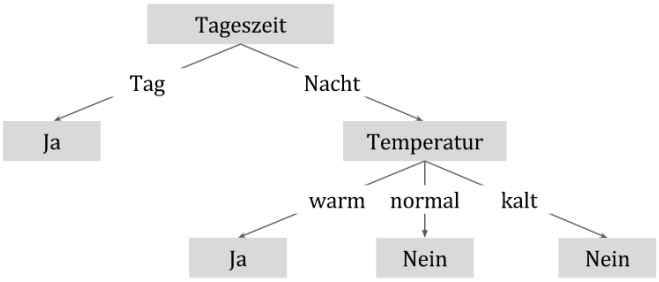


Abbildung 2.15: Entscheidungsbaum, der durch den ID3-Algorithmus für den Datensatz aus Beispiel 2.3 erzeugt wurde.

- `if` *Tageszeit* = *Nacht* `and` *Temperatur* = *warm* `then` *Spaß* = *Ja*
- `if` *Tageszeit* = *Nacht* `and` *Temperatur* = *normal* `then` *Spaß* = *Nein*
- `if` *Tageszeit* = *Nacht* `and` *Temperatur* = *kalt* `then` *Spaß* = *Nein*

Der Klassifikator, das heißt der Entscheidungsbaum, wird beim ID3 Algorithmus nach folgenden Muster erstellt: Der Baum wird Top-Down erzeugt, dass heisst beginnend bei der Wurzel bis zu den Blättern. Da in jedem Knoten genau ein Feature aufgespalten wird, wird an der Wurzel die Frage gestellt „*Welches Feature sollte zuerst getestet werden?*“. Um diese Frage zu beantworten, wird jedes Feature einem statistischen Test unterzogen und festzustellen, wie „gut“ es 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 erhält, die den jeweiligen Feature-Wert bestitzt. 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]

Das Wort „gut“ wurd ein dieser Beschreibung in Anführungsstrichen geschrieben, da es subjektiv ist und quantifiziert werden muss. Zur Quantifizierung der Information wird die Entropie nach Formel 2.20 als Hilfsmittel definiert. $p_i$ ist die Wahrscheinlichkeit, dass in einem Datensatz $D$ eine Instanz mit der Klasse $i \in C$ angetroffen wird.

$$H(p) = -\sum_{i \in C} p_i \cdot \log_2 p_i \tag{2.20}$$

Die Entropie quantifiziert die *Unreinheit des Datensatzes*. Angenommen, ein Datensatz hat zwei Klassen, $C = \{+, -\}$. Existiert der gesamte Datensatz nur aus einer der beiden Klasse, ist die Entropie $-p_+ \log_2 p_+ - p_- \log_2 p_- = -1log_21 - 0log_20 = 0$. Das heißt, dass die *Unreinheit des Datensatzes* 0 beträgt. Ist die *Unreinheit des Datensatzes* hingegen maximal, das heißt es liegen exakt 50% positive und 50% negative Samples vor, ist die Entropie $-p_+ \log_2 p_+ - p_- \log_2 p_- = -0.5log_{0.5}1 - 0.5log_20.5 = 1$. [32, S. 135]

Es ist das Attribut in einem Knoten zu wählen, welches den höchsten *Informatoinsgewinn* gewährleistet, das heißt, zu einer bestmöglichen *Reinheit* bei der alleinigen Unterteilung des Datensatzes auf Basis dieses Attributs führt. Der informationsgewinn eines Features $f$ für den Datensatz $D$ wird nach Formel 2.21 definiert. $v$ sind alle möglichen Werte dieses

Features. $|D|$ beschreibt die Anzahl an Instanzen des Datensatzes. $D_v$ ist die Untermenge an Instanzen, die für das Feature $f$ den Wert $v$ besitzen.[32, S. 136 - 137]

$$\text{Gain}(D, f) = H(D) - \sum_{v \in dom(f)} \frac{|D_v|}{|D|} H(D_v) \tag{2.21}$$

Für das Beispiel aus Tabelle 2.15 ergibt sich für den ersten Test folgende Berechnung des Informationsgewinnes der beiden Features *Temperatur* und *Tageszeit*. Da die Tageszeit den höheren Informationsgewinn gewährleistet, wird dieses Features in der Wurzel gewählt.

$$H(D) = -p_+ \log_2 p_+ - p_- \log_2 p_- = -\frac{4}{6}\log_2(\frac{4}{6}) - \frac{2}{6}\log_2(\frac{2}{6}) = 0.91 \tag{2.22}$$

$$\begin{aligned} Gain(D, Tageszeit) = 0.91 - \Big( & \overbrace{\frac{3}{6} \cdot (-\frac{3}{3}\log_2\frac{3}{3} - -\frac{0}{3}\log_2\frac{0}{3})}^{Tag} \\ & \underbrace{\frac{3}{6} \cdot (-\frac{1}{3}\log_2\frac{1}{3} - -\frac{2}{3}\log_2\frac{2}{3})}_{Nacht} \Big) = 0.86 \end{aligned} \tag{2.23}$$

$$\begin{aligned} Gain(D, Temperatur) = 0.91 - \Big( & \overbrace{\frac{2}{6} \cdot (-\frac{2}{2}\log_2\frac{2}{2} - -\frac{0}{2}\log_2\frac{0}{2})}^{warm} \\ & \underbrace{\frac{2}{6} \cdot (-\frac{1}{2}\log_2\frac{1}{2} - -\frac{1}{2}\log_2\frac{1}{2})}_{normal} \\ & \underbrace{\frac{2}{6} \cdot (-\frac{1}{2}\log_2\frac{1}{2} - -\frac{1}{2}\log_2\frac{1}{2})}_{kalt} \Big) = 0.66 \end{aligned} \tag{2.24}$$

Algorithmus1 zeigt den Ablauf des ID-3 in Pseudocode. $D$ ist die Menge aller Test-Examples, $X$ ist die Menge aller Features, $C$ ist die Menge aller Klassen, $f_{parent}$ das Feature des momentanen Eltern-Knotens und $v_{parent}$ der Wert des zum momentan konstruierten Knotens eingehenden Kante. [32, S. 139] [33, S. 56]

**Algorithm 1** ID3-Algorithmus in Pseudocode

```
1: tree = {}
2: function ID3(D, X, C, f_parent, v_parent )
3:                                   ▷ If all Examples have the same label, return a leaf with that Label
4:     if ∀e ∈ D : ∃k ∈ C : e.c = k then
5:         tree = tree ∪ {(f_parent, v_parrent, k)}
6:         return
7:     else
8:                                                ▷ If there are no Features left to test, return a leaf with
9:                                              ▷ the most common Label of the Examples remaining in D
10:        if isEmpty(X) then
11:            tree = tree ∪ {(f_parent, v_parrent, most common Label in D)}
12:            return
13:        else
14:                          ▷ Choose the feature that maximizes the Information-Gain to be the next node
15:            f_best = max_{f∈X} Gain(D, f)
16:                                                                  ▷ Add a Branch to this node
17:            tree = tree ∪ {(f_parent, v_parrent, f_best)}
18:                                                  ▷ Remove the feature from the set of features
19:            X_/f ← X/f_dom
20:            for v ∈ f_best do
21:                ▷ Calculate the new Dataset D_/f by removing all instances with the corresponding value
22:                D_/f ← ∀e ∈ D : e.f_best = v
23:                                                              ▷ Recursivly call the algorithm
24:                ID3(D_/f, X_/f, f_dom, v)
25:            end for
26:        end if
27:    end if
28: end function
```

Der ID3-Algorithmus hat folgende **Vorteile**:

**Kurze Entscheidungsbäume** Der Klassifizierer versucht, möglichst kurze Entscheidungsbäume zu bauen, indem Features mit hohem Informationsgewinn bevorzugt werden. Dies ist eine Umsetzung von *Ocam's Razor*: „Bevorzuge die kürzeste Hypothese“

**Verständlichkeit** Der Klassifikator ist für den Menschen verständlich, da er sich in Regeln übersetzen lässt (im Gegensatz zu zum Beispiel Neuronalen Netzen). Es existiert die unbewiesene Hypothese, dass der Mensch bei der Klassifizierung intuitiv ähnlich vorgeht wie der ID3-Algorithmus.[33, S. 63 - 65]

Der ID3-Algorithmus hat folgende **Nachteile**

**Nur Diskrete Werte** Der Algorithmus akzeptiert keine kontinuierlichen Werte [33, S. 72]

**Overfitting** Der Algorithmus neigt zu *Overfitting*. Overfitting bedeutet, dass der erzeugte Klassfikator $c$ zwar einen möglichst geringen Fehler in Bezug auf den *Trainings-Datensatz* hat, es jedoch einen anderen Klassifikator $c'$ gibt, welcher in Bezug auf 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 mehr 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. Es gibt keinen festen Beweis für das vorhandensein von Overfitting. Methoden zum Feststellen von Overfitting sind:

- Verwendung eines seperaten Test-Datensatzes, welcher bestätigt, dass der für den Trainings-Datensatz erzeugte Klassifikationsfehler auch bei bisher unbekannten Instanzen erzeugt wird.
- Verwendung von Statistischen Tests, die eine signifikante Reduktion des Klassifikationsfehlers bei Erweiterung des Entscheidungsbaumes beweisen.
- Expertenwissen über applikationstypischen Tiefen von Entscheidungsbäumen.[33, S. 66 - 70]

**Lokale Maxima** Der Algorithmus bevorzugt greedy Attribute, die zum Zeitpunkt der Berechnung den höchsten Inforamtionsgewinn gewährleisten. Dabei besteht die Gefahr, dass der Algorithmus in ein lokales Maximum läuft.[33, S. 66 - 70]

### 2.4.2 Gütemaße binärer Klassifikatoren

Ein binärer Klassifikation ist eine, bei dem es nur zwei Klassen gibt, das heißt $|C| = 2$. Applikationsabhängig werden die beiden Klassen als *Positive* und *Negative*, 1 und 0 oder *True* und *False* beschrieben. Eine Klassifikation, bei der ein tatsächliches Positive richtig 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]. Das System wird entsprechend für die Klassifikation tatsächlicher Negatives angewandt und ergibt. *True-Negatives* [TN] und *False-Positives* [FP]. Die *Confusion Matrix* in Abbildung 2.16 gibt eine Übersicht über die vier möglichen Klassifikations-Ergebnisse. [25, S. 213 - 214]

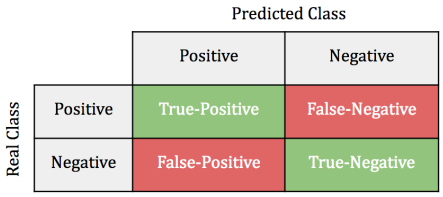


Abbildung 2.16: Confusion-Matrix (nach: [25, S. 214])

Die insgesamte Güte einer Klassifikation wird durch die *Accuracy* nach Formel 2.25 bestimmt. Eine Accuracy von 100% bedeutet, dass *alle* Instanzen richtig klassifiziert werden, eine Accuracy von 50% bedeutet, dass die Hälfte aller Instanzen richtig klassifiziert werden, was der Güte einer rein zufälligen Wahl entspricht. [25, S. 214]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.25}$$

Die Accuracy beziffert die insgesamte Performance des Klassifikators, 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 Accuracy von 50% beispielsweise dadurch entstehen, dass *alle* Instanzen als Positives markiert werden, also sowohl die Positives richtigerweise als Positives, aber die Negatives fälschlicherweise ebenfalls als Positives. Im Umgedrehten Fall ergibt die Klassifizierung aller Instanzen als Negatives ebenfalls eine Accuracy von 50%. In einem dritten Fall irrt sich die Klassifikator gleich oft bei der Einordnung der Negatives

und Positives. Die Maße *Sensitivity* und *Specificity* geben Aufschluss über die Güte der Klassifikation hinsichtlich der Positives und Negatives. Die *Sensitivity*, auch bezeichnet als *True-Positive-Rate*, bemisst den Anteil tatsächlicher Positives, die auch als solche erkannt wurden, nach Formel 2.26. Eine Sensitivity von 100% bedeutet, dass alle Positives durch den Klassifikator erkannt wurden. Die Erkennungsrate der Negatives hat keinen Einfluss auf die Sensitivity. Eine hohe Sensitivity lässt sich somit „einfach“ erzielen, in dem man *alle* Instanzen immer als Positives klassifiziert. Die Specificity nach Formel 2.27 bestimmt analog zur Sensitivity den Anteil der korrekt als Negatives bestimmten Instanzen. Ein Klassifikator, der alle Instanzen als Positives markiert, hat zwar eine Sensitivity von 100%, aber eine Specificity von 0%. Ergeben zwei verschiedene Klassifikationsmodelle sehr ähnliche Accuracies, hilft die Bestimmung der Sensitivity und Specificity bei der Auswahl des für den Anwendugnsfall Adäquteren Klassifikators. So ist beispielsweise bei der Bestimmung von schweren Krankheiten eventuell ein Klassifikator mit höherer Sensitivity wünschbar, um die Wahrscheinlichkeit zu minimieren, dass die entsprechende Krankheit nicht erkannt wird. [28] [25, S. 222]

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{2.26}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.27}$$

# 3 Konzept zur Visualisierung von Schmerz Scores aus akustischen Signalen

Das in dieser Arbeit umzusetzende System muss den folgenden Anforderungen genügen:

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 visualiseren.
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.

## 3.1 Literaturüberblick

In diesem Kapitel wird ein Überblick über bereits veröffentlichte Ansätze zur Analyse akustischer Signale von Neugeborenen gegeben.

Ein großer Teil der Veröffentlichungen stellt Algorithmen zur Klassifizierung einzelner Cry Units vor, entweder bezüglich der Weinursache (Hunger, Angst, Schmerz, ... ) oder zur Diagnose bestimmter Krankheiten. Diese Methoden sind in den meisten Fällen nicht für die kontinuierliche Analyse geeignet, sondern haben das Ziel, eine gegebenen Cry-Unit mit einer möglichst hohen Genauigkeit bezüglich der Ursache zu klassifizieren. Probleme wie Hintergrundrauschen, Berechnungsaufwand oder kontextuelle Informationen spielen eine untergeordnete Rolle. 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 Verarbeitungspipeline 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. Ana-

lysiert 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“ [42] 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 Schmerzindika-

toren 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 Systeme vorgestellt, deren Zielstellungen dem Thema dieser Masterarbeit ähneln. Keine der präsentierten Verarbeitungs-Pipelines 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. [42] [12].

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

1. **Vorverrbeitung** (engl. *Pre-Processing*) des Signals, beschrieben in Kapitel **??**.
2. **Voice Activity Detection**. Das Audiosignal wird in einander überlappende Zeitfenster weniger Millisekunden aufgeteilt. Mit Hilfe eines Klassifizierungsalgorithmus werden die Zeitfenster in als *stimmhaft* oder *nicht stimmhaft* markiert. Ununterbrochene Reihen stimmhafter Signalfenstern werden zu *Cry-Units* zusammengefasst, welche die Grundlage der darauf folgenden Verarbeitungsschritte bilden. Diese Idee wurde aus der Dissertation von Várallyay [51, S. 16 - 17] übernommen, welcher Cry-Units als *Segmente* bezeichnet. Die Voice Activity Detection wird in Kapitel 4 vorgestellt.
3. **Segmentierung** (engl *Segmenting*). Die Cry-Units werden zu Segmenten zusammengefasst. Dieser Schritt ist notwendig, weil die Ableitung des Schmerz Score nicht aus den Informationen einer Cry-Unit, sondern aus dem Verbund mehrerer Cry-Units geschieht. Keine der in Kapitel 3.1 vorgestellten Veröffentlichungen beschreibt ein Verfahren, welches adaptiert werden könnte. Daher wird ein simpler Algorithmus für die Segmentierung vorgeschlagen, welcher für eine kontinuierliche Auswertung implementiert werden kann. Die Segmentierung wird in Kapitel 5.1 vorgestellt.
4. **Extrahierung von Eigenschaften** (engl. *Feature Extraction*), das heißt die Berechnung von Features für jedes Segment, aus denen der Schmerz Score abgeleitet werden kann. Dieser Prozess wird in Kapitel 5.2.1 vorgestellt.
5. **Ableitung der Schmerz Score** (engl. *Prediction of Pain Score*) aus den Eigenschaften der Segmente. Dieses Problem kann entweder als Klassifizierungs- oder Regressionsaufgabe modelliert werden. Die grundlegende Idee wird in Kapitel 5.2 vorgestellt und in Kapitel 5.2.2 weiter ausgearbeitet.
6. **Visualisierung** (engl. *Visualisation*) der abgeleiteten Schmerz Score. In dieser Arbeit werden mehrere Varianten vorgeschlagen, welche die Höhe des Schmerz Score in seinem zeitlichen Verlauf auf Ampelfarben abbildet. Die Visualisierung wird in Kapitel **??** vorgestellt.

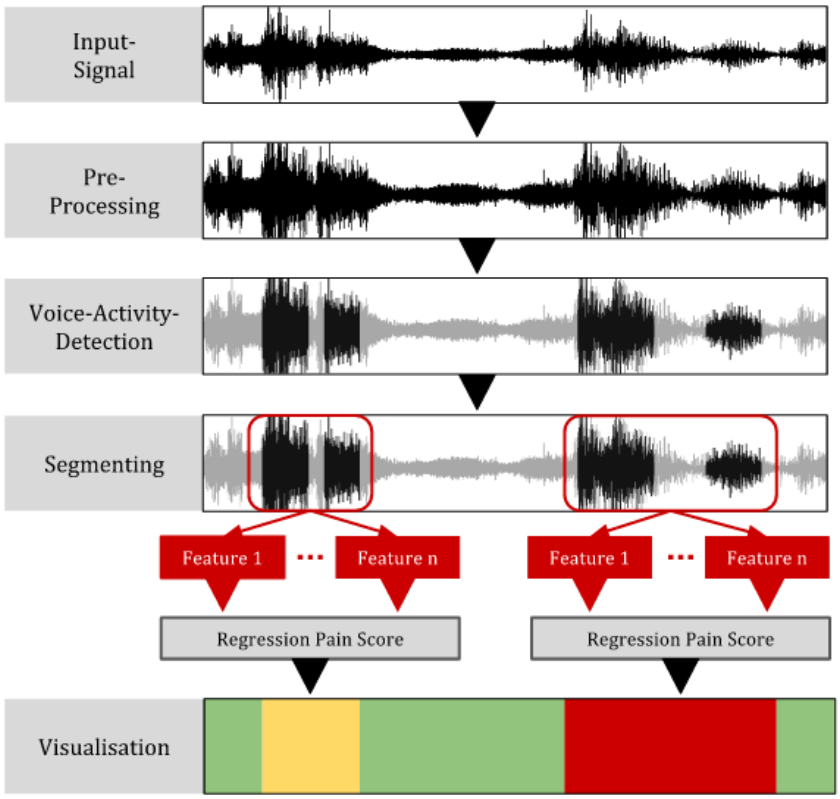


Abbildung 3.1: Überblick über die Verarbeitungs-Pipeline dieser Arbeit

# 4 Voice Activity Detection

Das Ziel ist, in einem Audiosignal diejenigen Stellen zu markieren, in denen Stimme enthalten ist. Abbilung 4.1 visualisiert ein Beispiel für eine solche Markierung. Zu sehen ist der Zeitbereich eines Audiosignales mit fünf klar erkennbaren Cry-Units. Die rote Linie, die das Signal überspannt, bildet die Zeiteinheiten des Eingangssignales in die binären Kategorien 1*hat*= *stimmhaft* (engl. *voiced*) und 0*hat*= *Stille* (oder *nicht-stimmhaft*, engl. *not voiced*) ab.

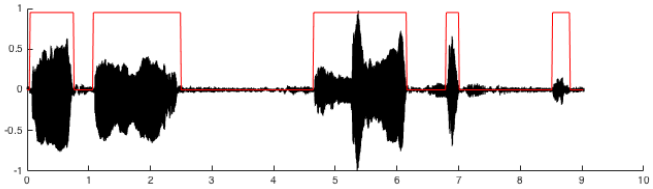


Abbildung 4.1: Markierung stimmhafter Bereiche in einem Audiosignal. Schwarz: Das Eingangssignal $x[\ ]$. Rot: Klassifizierung in stimmhaft/Stille. Es sind fünf Cry-Units zu erkennen.

Die Erkennung des Vorhandenseins von Stimme in einem Signal wird als *Voice Activity Detection (VAD)* oder auch *Speech Detection* bezeichnet. Das Ziel ist die Unterscheidung von denjenigen Zeiträumen im Signal, in denen Stimme enthalten ist, von den Zeiträumen ohne Stimme. Die größte Herausforderung für VAD-Algorithmen ist die robuste Erkennung bei Signalen mit Rauschen unbekannter Stärke und Natur. [22, S. 1] [50, S. 1]

Der Grundlegende Aufbau eines VAD-Algorithmus ist wie folgt.

1. **Windowing:** Unterteilung des Signals in (einander überlappende) Signalfenster.
2. **Extraktion von Eigenschaften** aus den einzelnen Signalfenstern.
3. **Entscheidung** über die Präsens oder Abwesenheit von Stimme für jedes Signalfenster auf Grundlage der extrahierten Attribute mit Hilfe von Entscheidungsregeln wie Grenzwerten.
4. **Decision-Smoothing**, das nachträgliche Hinzufügen oder Entfernen von Entscheidungen mit Hilfe von kontextuellen Informationen der umliegenden Entschiedungen.[20, S. 8 - 9] [22, S. 1 - 2]

Auch die in dieser Arbeit durchgeführte Voice Activity Detection folgt diesem Schema. Das Windowing wird in Kapitel 4.0.1, die Extraktion von Eigenschaften in Kapitel 4.0.2, die Entscheidung in Kapitel 4.0.3 und das Decision-Smoothing in Kapitel 4.0.5 beschrieben. Es wurden Ideen verwendet, 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.

### 4.0.1 Windowing

Wird die Voice Activity Detection für ein Signal $x[\,]$ durchgeführt, wird dieses zuerst nach dem in Kapitel 2.2.4 beschriebenen Verfahren nach Gleichung 2.15 in die Signalfenster $x_0[\,], \ldots, x_m[\,]$ zerlegt. Der Prozess wird als „Windowing“ bezeichnet. Die Signalfenster 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.

### 4.0.2 Extraktion von Eigenschaften

Um die Entscheidung zu treffen, ob in einem beim Windowing enstandenen Signalfenster $x_i[\,]$ Stimme enthalten ist, wird zunächst eine Reihe an Eigenschaften für das Signalfenster berechnet. Auf Basis dieser Eigenschaften kann daraufhin die Entscheidung gefällt werden. Einer der primären Forschungsgegenstände der VAD ist die Erprobung und Evaluation von Eigenschaften zu diesem Zweck. In diesem Kapitel wird eine Reihe an Attribute vorgestellt, mit dem Ziel, diejenigen zu identifizieren, die sich im Zusammenhang mit kindlichen Lautäußerungen am besten zur Erkennung von Stimme eignen. Das Vorgehen ist folgendermaßen:

1. Es wurde ein Testdatensatz mit Audioaufnahmen von kindlichen Lautäußerungen erstellt. Diese Datensätze werden in Kapitel 4.0.3 beschrieben.
2. Jedes Signal des Datensatzes wurde nach dem eben beschriebenen Vorgehen vorverarbeitet und daraufhin in kürzere Signalfenster zerlegt. Für jedes Signalfenster wurden die Eigenschaften berechnet, die in den folgenden Kapiteln 4.0.2 bis 4.0.2 beschrieben werden.
3. Das Ziel war es, eine möglichst kleine Untermenge an Eigenschaften zu finden, auf deren Basis die Entscheidung über das Vorhandensein von Stimme mit einer möglichst hohen Genauigkeit durchgeführt werden kann. In Kapitel 4.0.3 wird beschrieben, wie die verschiedenen Eigenschaften bezüglich ihrer Performance evaluiert wurden.

Für jedes Signalfenster $x_i[\,]$ à 25 ms des Testdatensatzes wurden die folgenden Features aus den Kategorien **Zeitbereich**, **Frequenzbereich**, **Cesptrum** und **Autokorrelation** erprobt.

#### Zeitbereich

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

Moattar et al. [30] bezeichnen den Energiegehalt eines Signals als das für die VAD am häufigsten angewandte Attribut. Daher wurde der RMS-Wert nach Gleichung 2.7 für die Signalfenster berechnet. Hintergrund ist, dass der Energiegehalt eines Stimmsignals typischerweise höher ist als der des Hintergrundrauschens. Bei geringem Signal/Rauschabständen ist diese Bedingung jedoch nicht immer gegeben. Als zweites Attribut des Zeitbereiches wurde die *Zero Crossing Rate* berechnet. Die ZCR nach Formel 4.1 gibt an, wie häufig ein Vorzeichenwechsel im Signal vorkommt. Eine höhere ZCR weist auf Stille 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.

$$\mathrm{ZCR}(x_i[\,]) = \sum_{0}^{N-1} |\mathrm{sng}(x_i[n]) - \mathrm{sng}(x_i[n-1])| \tag{4.1}$$

### Autokorrelation

Neben den in Kapitel 4.0.2 genannten „einfachen“ Attributen des Zeitbereiches wurde die Autokorrelation erprobt. Wie in Kapitel 2.2.5 ausgeführt, weisen stimmhafte Signale eine tendenziell stärkeres periodisches Verhalten als das Hintergrundrauschen auf. Daher eignet sich die in Kapitel **??** vorgestellte Autokorrelation, um diese Periodizität festzustellen. Es wurden die Attribute *Maximum Autocorrelation Peak* [$aMax$] und (*Autocorrelation Peak Count*) [$aCount$] berechnet.

Beide Eigenschaften wurden von Kristjansson et al. [50, S. 1 - 2] zur VAD beschrieben. Die (*Maximum Autocorrelation Peak*) wird in Formel 4.2 definiert und bestimmt die höchste Magnitude im Autokorrelationssignal. Eine hoher [$aMax$]-Wert weist auf eine starke Periodizität hin. Das zweite Attribut ist die *Autocorrelation Peak Count* nach Formel 4.3. Dabei wird die Anzahl an Signalspitzen im Autokorrelationssignal gezählt. Rauschen erzeugt höhere [$aCount$]-Wert als stimmhafte Signale, bedingt durch die vielen zufällig Verteilten Periodizitäten. Aus Kapitel 2.3.1 ging hervor, dass die Grundfrequenz der Stimme von Neugeborenen zwischen 200 und 2000 Hz liegt, weshalb auch nur in Lags dieses Bereichs verwendet wurden.

$$\mathrm{aMax}(x_i[\,]) = \max_k \mathrm{mag}\{\text{NA-Corr}_k(x_i[\,])\} \tag{4.2}$$

$$\mathrm{aCount}(x_i[\,]) = \operatorname*{count}_k \mathrm{mag}\{\text{NA-Corr}_k(x_i[\,])\} \tag{4.3}$$

### Frequenzbereich

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

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 4.5 berechnet. Das Signal $px_i[\,]$ ergibt sich durch die Normalisierung des $N$-Punkte langen Spektrums nach Formel 4.4. Neben der von Kristjansson et al. [50] vorgestellten normalisierten spektralen Entropie wurde zusätzlich die *unnormalisierte Spektrale Entropie* nach Formel 4.6 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. Bei der normalisierten spektralen Entropie ist zu erwarten, dass Frequenzfenster ohne Stimme einen höheren Wert aufweisen als Fenster mit Stimme. Bei der unnormalisierten spektralen Entropie ist zu erwarten, dass Signalfenster mit Stimme einen höheren Wert aufweisen als Signale mit Stille.[1]

In die Berechnungen wurden nur die Frequenzen im Bereich von 200 - 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.

$$px_i[n] = \frac{X_i[n]}{\sum_{k=1}^{N} X_i[k]} \tag{4.4}$$

$$\mathrm{SEnt}_n(px_i[\,]) = -\sum_{k=1}^{N} px_i[k] \cdot \log(px_i[k]) \tag{4.5}$$

$$\mathrm{SEnt}_u(X_i[\,]) = -\sum_{k=1}^{N} X_i[k] \cdot \log(X_i[k]) \tag{4.6}$$

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 4.7 berechnet, welche die höchste Amplitude hat. Es wird dabei, im Gegensatz zur spektralen Entropie, der gesamte Frequenzraum betrachtet. Ein stimmhaftes Signal hat typischerweise eine höhere $f_{dom}$ als ein stimmloses Signal, bedingt durch die hohe Amplitude der Grundfrequenz.

$$f_{dom}(X_i[\,]) = \arg\max\{X_i[\,]\} \tag{4.7}$$

### Cepstrum

In Kapitel **??** wurde das Cepstrum vorgestellt und erläutert, wie Peaks im oberen Quefrency-Bereich auf das Vorhandensein eines periodischen, obertonreichen Signals, wie zum Beispiel Stimme, hinweisen. Aus dem Cepstrum-Bereich wurden die Features *Upper Cepstrum Peak* $[Ceps_{mag}]$ und *Upper Cepstrum Peak Location* $[Ceps_{loc}]$ berechnet.

Ahmadi et al. [46] sowie Kristjansson et al.[50] schlagen vor, die höchste Magnitude im oberen Quefrency-Bereich (Upper Cepstrum Peak) als Feature zu verwenden. Formel 4.8 definiert die Berechnung. $c_i[\,]$ ist das Cepstrum des $i$-ten Frequenzfensters $X_i[\,]$. Wie in

[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!

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 4.8 nur Quefrency-Werte in diesem Bereich betrachtet. Ein hoher $Ceps_{mag}$-Wert weist auf das Vorhandensein von Stimme hin. Als zweites Attribut wird die Quefrency der höchsten Amplitude des Cepstrum (Upper Cepstrum Peak Location) nach Formel 4.9 berechnet. Bei Signalfenstern mit Stille ist es wahrscheinlicher, dass sich die höchste Amplitude am Mindest- oder Maximalwert des durchsuchten Quefrency-Bereiches befindet.

$$Ceps_{mag}(c_i[\,]) = \max \mathrm{mag}\{c[\,]\} \tag{4.8}$$

$$Ceps_{loc}(c_i\,) = \arg\max\{c[\,]\} \tag{4.9}$$

Abbildung 4.2 visualisiert alle vorgestellten Attribute, die für die Voice Activity Detection erprobt 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 $1\hat{=}$ *stimmhaft* und $0\hat{=}$ *nicht stimmhaft*. Alle darunter liegenden Plots zeigen den zeitlichen Verlauf der entsprechenden Attribute.

#### Konstruktion des Feature-Raumes

Abbildung 4.3 zeigt in (A) des zeitlichen Verlauf des *RMS*-Attributes eines Signals mit einem Signal/Rauschabstand von 50 dB. Die Zeiträume mit Stille 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.

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 zunächst noch keine Stimme enthalten, und der Feature-Wert des Rauschens somit anhand dieser Fenster bestimmt werden kann. Bei einer langanhaltenden und kontinuierlichen Analyse können sich sowohl die Signal/Rauschabstände 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 ein Zeitfenster mit Stille entsteht. Zeskind et al. [41, S. 325] haben diesen Wert mit $t_{max} = 4.75\,\mathrm{s}$ 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 wird das *Differenz-Feature* $\mathrm{Diff_t}(\mathrm{Feat}(x_i[\,]))$ nach Formel 4.10 definiert als die Differenz zwischen dem 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 Vergangen-

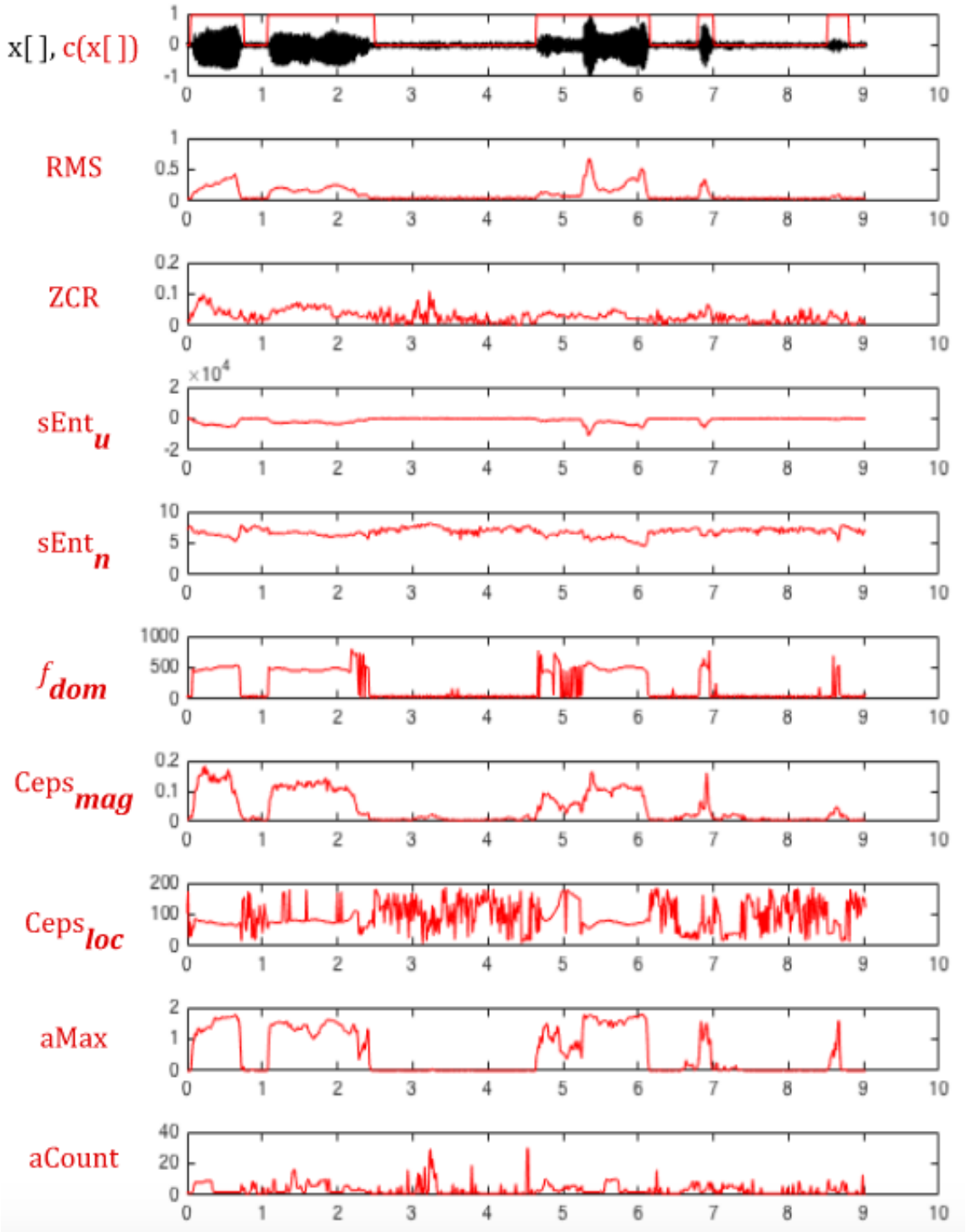


Abbildung 4.2: Übersicht über alle Features, die für die Voice Activity Detection erprobt wurden.

heit zu durchsuchende Zeitbereich in Sekunden $> t_{max}$. In Abbildung 4.3 wird in (C) das Differenz-Feature für den RMS-Wertes gezeigt.

$$\text{Diff}_t(\text{Feat}(x_i[\,])) = Feat(x_i[\,]) \; - \min_{k=i-z\ldots i}\{Feat(x_k[\,])\}, \qquad z = \frac{2 \cdot t}{t_{xi}} \tag{4.10}$$

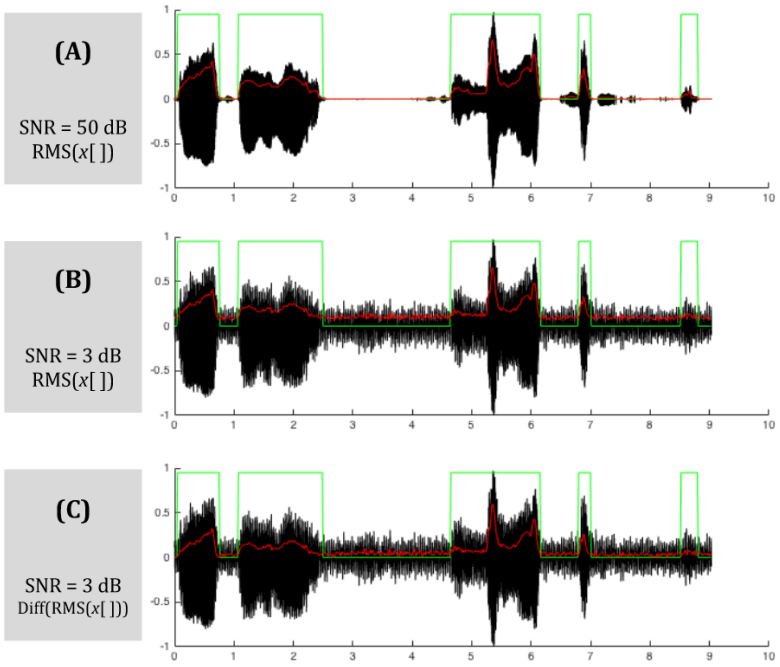


Abbildung 4.3: Das RMS-Feature bei verschiedenen Signal/Rausch-Abständen. Schwarz: Eingangs-Signal $x[\,]$. Grün: Klassifizierung in Stimmhaft/Stille. Rot: Feature-Wert.

Der Feature-Raum wurde schlussendlich folgendermaßen zusammengesetzt: Die ersten 9 Features bildeten die in Attribute *RMS, ZCR, SEnt*$_u$*, SEnt*$_n$*,* $f_{dom}$*, Ceps*$_{mag}$*, Ceps*$_{loc}$*, aMax* und *aCount*. Weiterhin wurde für jedes Attribut nach Formel 4.10 das Differenz-Feature mit $t = 5\,\mathrm{s}$ berechnet. Die Features *ZCR, SEnt*$_u$ und *aCount* wurden vor der Berechnung des Differenz-Features bezüglich ihres Vorzeichens invertiert, da bei Ihnen ein niedriger anstatt ein hoher Wert stimmhafte Signale anzeigen. Das einzige Attribut, für den kein Differenz-Feature berechnet wurde, ist das *Ceps*$_{loc}$-Attribut, da es bei Stille sowohl einen höheren als auch einen niedrigeren Wert annehmen kann. Der Feature-Raum umfasste somit insgesamt $9 + 8 = 17$ Dimensionen. Gleichung 4.11 verdeutlicht die Zusammensetzung des Feature-Vektors $v_i$, der für das Signalfenster $x_i[\,]$ berechnet wurde.

$$v_i = \Big(\mathrm{RMS}(x_i[\,]), ...,\ \mathrm{aCount}(x_i[\,]), \mathrm{Diff}_t(\mathrm{RMS}(x_i[\,]))....\mathrm{Diff}_t(-\ \mathrm{aCount}(x_i[\,]))\Big) \quad (4.11)$$

### 4.0.3 Thresholding

#### Finden der Grenzwerte

Das Ziel war es nun, Grenzwerte für die Attribute zu finden, bei deren Über- oder Unterschreitung das jeweilige Signalfenster als *stimmhaft* klassifiziert wird. Abbildung 4.4 verdeutlicht das Prinzip für das *RMS*-Attribut. Diese Entscheidung nach einem Grenzwert ist ein klassisches Vorgehen bei der Voice-Activity-Detection. Eine binäre Klassifizierung nach dem Muster $C(x_i[\,]) = \{1, \text{wenn } \mathrm{RMS}(x_i[\,]) \geq 0.18, \quad 0 \text{ sonst}\}$ würde in diesem Fall eine weitestgehend richtige Klassifizierung vornehmen.

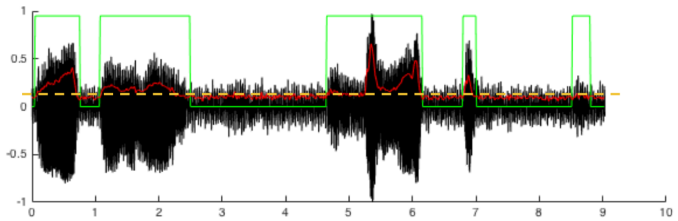


Abbildung 4.4: Thresholding eines Features. Schwarz: Das Eingangssignal $x[\,]$. Grün: Klassifizierung in Stimmhaft/Stille. Rot: RMS-Feature. Orange: Grenzwert

Eine Methode zum Finden der optimalen Grenzwerte ist der in Kapitel **??** vorgestellte *C4.5*-Algorithmus. Da der *C4.5*-Algorithmus Entscheidungsbäume erstellt, kann die Entscheidung über das Vorhandensein von Stimme in einem Signalfenster aufgrund der Verkettung von Grenzwerten mehrerer Attribute in Folge gefällt werden. Ein Beispiel wird in Listing 4.1 dargestellt, bei dem die Klasse eines Signalfensters hierarchisch zuerst nach einem Grenzwert für $\text{Ceps}_{mag}$ und danach für den RMS-Wert entschieden wird.

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
```

### Trainings- und Testdatensätze

Es wurden sechs Audioaufnahmen mit Weinen verschiedener Babies von der freien Online-Sound-Bibliothek `https://www.freesound.org/` heruntergeladen und zu Segmenten à 10 s beschnitten. Es handelt sich um weitgesgehend rauschfreie Aufnahmen, die von verschiedenen Babys stammen. In den Audiosignalen wurden manuell die Zeitbereiche markiert, welche Stimme enthalten. Es wurden *keine* Geräusche markiert, bei denen es sich offensichtlich um Einatumungs-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. 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 spielt eine Sonderrolle, da es nur zur Verifikation verwendet wird. So wurden vier Mengen an Audiosignalen erzeugt:

$\mathbf{A}_{50\,\text{dB}}$ enthält $3 \cdot 6 = 18$ Audiosignale, bei dem alle sechs Baby-Aufnahmen mit den drei Rauschsignalen bei einem Signal/Rausch-Abstand von 50 dB überlagert wurden

$\mathbf{A}_{3\,\text{dB}}$ enthält $3 \cdot 6 = 18$ Audiosignale, bei dem alle sechs Aufnahmen der Babys mit den drei Rauschsignalen bei einem Signal/Rausch-Abstand von 3 dB überlagert wurden

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

$\mathbf{A}_{7\,\mathrm{dB}*}$ enthält 1 Audiosignal, bei dem eine siebte Aufnahme eines Babys mit einem vierten Rauschsignal bei einem Signal/Rausch-Abstand von 7 dB überlagert wurde

Im nächsten Schritt werden die eigentlichen Datensätze $D_{SNR,Feats}$ gebildet, in dem Audiosignale dieser Signalmengen (1) wie in Kapitel **??** beschrieben vorverarbeitet werden, (2) wie in Kapitel 4.0.1 in die Signalfenster à 25 ms zerlegt werden und (3) für jedes Signalfenster der durch Gleichung 4.11 definierte Featurevektoren berechnet wird. Außerdem wird jedem Featurevektor die Klasseninformation *Stimme/Stille* zugewiesen.

Es ist rechnerisch zu aufwendig, alle genannten Features in einem kontinuierlichen System zur Voice Activity Detection zu berechnen. Daher werden die Datensätze in Untermengen bezüglich der verwendeten Features eingeteilt. Das Ziel ist es, eine möglichst kleine Untermenge an Features zu finden, die sich am besten für die Voice Activity Detection sowohl bei niedrigem als auch bei starkem Hintergrundrauschen eignet. Die Untermengen werden in Bezug auf die Methode gebildet, durch die die Features berechnet werden. Das heißt, dass beispielsweise die Untermenge *Zeit* die in Kapitel 4.0.2 beschriebenen Features *RMS* und *ZCR* sowie die dazugehören Differenzfeatures *Diff$_t$(RMS)* und *Diff$_t$(ZCR)* beinhaltet.

Die 9 Untermengen sind: { Zeitbereich, Frequenzbereich, Cepstrum, Autokorrelation, Zeit + Frequenzbereich, Zeit + Cepstrum, Zeit + Autokorrelation, Frequenzbereich + Cepstrum, Frequenzbereich + Autokorrelation }. Cepstrum- und Autokorrelation werden nicht gemeinsam in eine Untermenge hinzugefügt, da sie in Bezug auf den Berechnungsaufwand die aufwendigsten sind. So enthält beispielsweise der Datensatz $D_{3\,\mathrm{dB},Zeit}$ die Featurevektoren des Zeitbereiches für die Audiosignale mit einem Signal-Rausch-Abstand von 3 dB. Alle Audiosignal-Mengen [$\mathrm{A}_{50\,\mathrm{dB}}$], [$\mathrm{A}_{3\,\mathrm{dB}}$], [$\mathrm{A}_{50+3\,\mathrm{dB}}$] und [$\mathrm{A}_{7\,\mathrm{dB}}$] wurden in Datensätze umgewandelt. Es wurden schlussendlich $4 \cdot 9 = 36$ Datensätze gebildet.

### Training

Das Ziel ist, mit Hilfe des *C4.5*-Algorithmus einen Entschidungsbaum zu finden, der auf Basis einer möglichst geringen Feature-Menge eine möglichst hohe Klassifkationsgenauigkeit für sowohl niedrige als auch hohe Signal/Rausch-Abstänge erzielt. Die Frage ist, ob ein Entschiedungsbaum, der auf Basis von Signalen mit niedrigem SNR gebildet wird, auch für hohe SNRs eine hohe Klassifikationsgenauigkeiten erzielt, oder ob der umgedrehte Fall zutreffend ist. Daher werden die Entscheidungsbäume sowohl auf Basis verschiedener SNRs als auch verschiedener Feature-Untermengen gebildet. Die Entschäudungsbäume werden daraufhin gegen die Signale mit den verschiedenen SNRs evaluiert. Wird also beispielsweise der Datensatz $D_{50\,\mathrm{dB},Zeit}$ zum Training und der Datensatz $D_{3\,\mathrm{dB}}$ zum Testing verwendet, so wird berechnet, wie gut sich der Klassifikator unter Verwendung der Zeit-Features zur Klassifizierung niedriger SNRs eignet, obwohl er für hohe SNRs entworfen wurde. Dabei ist unerheblich, welche Features der Test-Datensatz verwendet, da bei der Evaluation nur die Klasseninformation der Instanzen verwendet werden.

Die Implementierung, die für den *C4.5*-Algorithmus verwendet wurde, ist der *REPTree* [2] der Open Source Data-Mining-Bibliothek *Weka*[3]. 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.

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

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

Es wurden insgesamt $3 \cdot 9 = 27$ Trainings-Datensätze erzeugt ( [3 SNR-Werte: 3 dB, 50 dB und 50+3 dB ] $\times$ [9 Feature-Untermengen]. Der Datensatz mit einem SNR von 7 dB wurde *nicht* zum Training verwendet). Mit diesen 27 Trainingsdatensätze wurden mit Hilfe des *REPTree*-Algorithmus 27 Klassifikationsbäume erzeugt. Jeder Klassifikationsbaum wurde gegen die 3 Testdatensätze $D_{3\,dB}$, $D_{50\,dB}$ und $D_{7\,dB^*}$ evaluiert und die Accuracy berechnet. Das Signal $A_{7\,dB^*}$ erfüllt dabei eine Sonderrolle, da es nicht in den Trainingsdatenstäzen enthalten ist und somit der Kontrolle bezüglich Overfitting dient. Da jeder Datensatz ungefähr dreimal so viele stimmhafte Examples wie nicht-stimmhafte enhthielt, wurde jede stimmlose Instanz eines Datensatzes dreimal eingefügt. Somit wurde in jedem Datensatz ein ausgewogenes Verhältnis zwischen positiven und negativen Examples gewährleistet. Um die Komplexitiät des Entscheidungsbaumes zu verringern eine Nutzung von möglichst wenig Features zur Klassifizierung zu erzwingen, wurde die maximale Tiefe des REPTree auf 2 gesetzt.

### Ergebnis

Die Evaluations-Ergebnisse sind in Tabelle .1 zu sehen. Für jeden Trainingsdatensatz mit einem bestimmten SNR und einer Feature-Untermenge wird die Accuracy für den jeweilgen Test-Datensatz mit einem SNR von 3 dB, 50 dB und 7 dB* angegeben.[4]. Außerdem wird der Durchschnittswert aller drei jeweiligen Accuracy-Werte angegeben.

Die Features, welche zu den höchsten Accuracy-Werten führten, sind die des *Cepstrum*-Bereiches, genauer gesagt das $\mathrm{Diff_t(Ceps_{mag})}$-Feature, da es vom REPTree als einziges Feature dieses Bereiches für die Entscheidungsbäume ausgewählt wurde. Die Entscheidungsbäume, die mit dem $\mathrm{Diff_t(Ceps_{mag})}$-Feature entworfen wurden, erreichten eine durchschnittliche Accuracy von mindestens 91,45%. Der nächstbeste Klassifikator mit einer Accuracy von 86,96% wurde unter Verwendung der Features des Zeitbereiches und der Autokorrelation auf dem Datensatz $D_{50+3\,dB,Zeit+Correlation}$ entworfen. Sobald die Cepstrum-Features in Verbindung mit den Features anderer Bereiche verwendet wurden, wurde das $\mathrm{Diff_t(Ceps_{mag})}$-Feature vom REPTree-Algorithmus bevorzugt und die Features der anderen Bereiche nicht mehr verwendet.

Auf Basis der Datensätze $D_{3\,dB,Ceps}$, $D_{3\,dB,Zeit+Ceps}$, $D_{3\,dB,Freq+Ceps}$, $D_{50+3\,dB,Ceps}$, $D_{50+3\,dB,Zeit+Ceps}$ sowie $D_{50+3\,dB,Freq+Ceps}$ wurde der selbe Klassifikator erzeugt, der in Gleichung 4.12 definiert wird. Wie zu sehen ist, handelt es sich um einen einfachen Grenzwert des $v.Diff_t(Ceps_{mag})$-Features, da trotz der höchst möglichen Baumtiefe von 2 nur eine Tiefe von 1 genutzt wurde.

$$C(v) = \begin{cases} 1, & \text{if } v.Diff_t(Ceps_{mag}) > 0.02, \\ 0 & \text{else} \end{cases} \tag{4.12}$$

Auf Basis der Datensätze $D_{50\,dB,Ceps}$ und $D_{50\,dB,Zeit+Ceps}$ wurde der Klassifikator nach

[4]Der Stern verdeutlicht die Sonderrolle des Datensatzes mit einem SNR von 7 dB, da er nur zu Evaluation verwendet wurde

Gleichung 4.13 erzeugt. Er unterscheidet sich von dem Klassifikator aus Gleichung 4.12 nur durch den Grenzwert.

$$C(v) = \begin{cases} 1, & \text{if } v.Diff_t(Ceps_{mag}) > 0.03, \\ 0 & \text{else} \end{cases} \tag{4.13}$$

Da der Klassifikator aus Gleichung 4.12 eine durchschnittliche Accuracy von 92,22% und der Klassifikator aus Gleichung 4.13 eine unwesentlich geringere Accuracy von 91,45% erzielt, wurden für beide Modelle die Specificity und Sensitivity berechnet, um eine Entscheidung für eines der beiden Modelle fällen zu können. Dazu wurden die Signalmengen $A_{3\,\mathrm{dB}}$, $A_{50\,\mathrm{dB}}$ und $A_{7\,\mathrm{dB}*}$ in Frames à 100 Windows zerlegt und für jedes Zeitfenster die Senstivity, Specificity und Accuracy 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 Specificity, aber geringere Senstivitiy 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 Senstivity 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 Klassifikations-Funktion eines Signalfensters $C(x_i[\,]) \mapsto \{0\hat{=}\text{Stille}, 1\hat{=}\text{Stimme}\}$ wird somit durch Gleichung 4.14 gegeben, wobei $c_i[\,]$ das Cepstrum des Signalfensters ist.

$$C(x_i[\,]) = \begin{cases} 1, & \text{if } v.Diff_t(Ceps_{mag}(c_i[\,])) > 0.02, \\ 0 & \text{else} \end{cases} \tag{4.14}$$

### 4.0.4 Markierung der Cry-Units

Wird die Voice-Activity-Detection für das Signal $x[\,]$ nach Gleichung 4.14 durchgeführt, ist das Ergebnis eine Zuordnung der Signalfenster $x_0[\,] \dots x_m[\,]$ zu den Klassen Stimme/Stille. Varallyay [51, S. 16 - 17] stellt die Idee vor, auf Grundlage der Informationen der Voice-Activity-Detection die Anfangs- und Endzeitpunkte der Cry-Units zu markieren (welche er als Cry-Segmente beschreibt). 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 4.5 veranschaulicht diese Gruppierung.

Formel 4.15 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 $windows = [x_0[\,], \dots, x_n][\,]$.

$$CU = (windows = \left[x_0[\,], \dots, x_n[\,]\right], start \in Zeit, end \in Zeit) \tag{4.15}$$

Die Dauer eine Cry-Unit $cu \in CU$ wird nach Formel 4.16 berechnet und mit $\lambda$ bezeichnet.

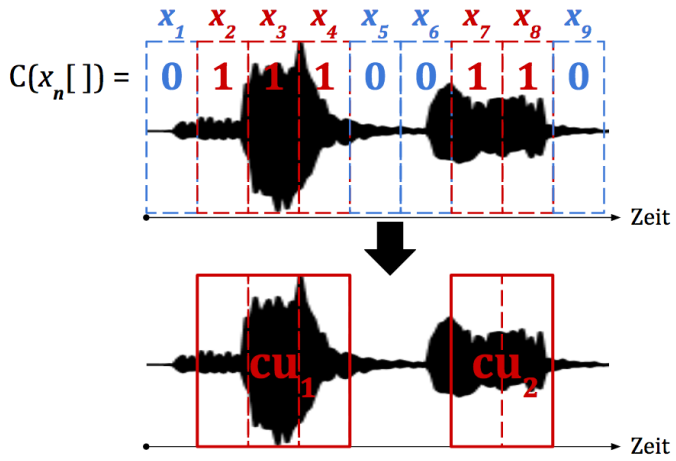


Abbildung 4.5: Zusammenfassung klassifizierter Signalfenster zu Cry-Units

Die zeitliche Dauer der Pause zwischen zwei Cry-Units $\mathrm{d}(cu_i, cu_j)$, wird nach Formel 4.17 berechnet. Diese Zusammenhänge werden in Abbildung 4.6 visualisiert.[24, S. 2]

$$\lambda(cu) = cu.end - cu.start \tag{4.16}$$

$$\mathrm{d}(cu_i, cu_j) = cu_j.start - cu_i.end \tag{4.17}$$

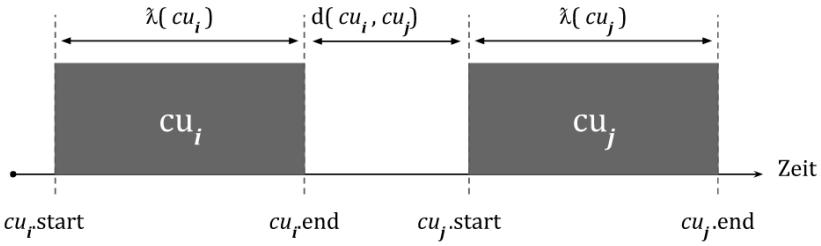


Abbildung 4.6: Beziehung zwischen agrenzenden Cry-Units, nach [24, S. 2]

Algorithmus 2 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 4.14. Die Funktion getTimeOf($x_i[\,]$) liefert die Anfangszeitpunkt des Signalfensters $x_i[\,]$.

### 4.0.5 Decision Smoothing

Abbildung 4.7 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.

**Algorithm 2** 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
```

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.

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: $\lambda_{min}$, die Mindestlänge einer akzeptierten Cry-Unit, und $d_{min}$, die Mindestlänge eines akzeptierten Stille-Segmentes. Das Decision-Smoothing wird nach den folgenden Entscheidungsregeln durchgeführt:

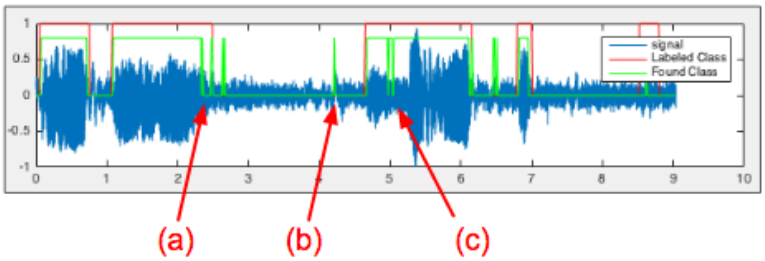


Abbildung 4.7: Klassifizierung vor dem Decision Smoothing

- 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 3 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 2 entstanden ist, sowie die Grenzwerte $\lambda_{min}, d_{min}$. Der Output der Funktion ist die Liste aller Cry-Units nach dem Decision-Smoothing $CU_{smoothed}$.

Abbildung 4.8 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 [41, S. 325] und beträgt 60 ms, welcher für $\lambda_{min}$ 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.

### 4.0.6 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 dieser grundlegenden Herangehensweise zur Voice Activity Detection kontextuelle Informationen in Bezug auf den zeitlichen Verlauf der Stimme jedoch nur in einem geringen

**Algorithm 3** Decision-Smoothing for VAD

**function** DECISIONSMOOTHING($CU_{all}, \lambda_{min}, d_{min}$)
  $CU_{smoothed} \leftarrow [CU_{all}[0]]$
  ▷ start for-loop at the *second* cry-Unit!
  **for** $i = 1, \ldots, length(CU_{all}) - 1$ **do**
    $cu_i \leftarrow CU_{all}[i]$
    $cu_{i-1} \leftarrow CU_{smoothed}[end]$
    **if** $\lambda(cu_i) > \lambda_{min}$ **then**
      ▷ Accept Cry-Unit
      **if** $\mathrm{d}(cu_{i-1}, cu_i) > d_{min}$ **then**
        $CU_{smoothed} \leftarrow [CU_{smoothed}, cu_i]$
      **else**
        ▷ Erase False-Negative Type (c)
        $cu_i \leftarrow \text{vereinige}(cu_i, cu_{i-1})$
        $CU_{smoothed} \leftarrow [CU_{smoothed}[1 : end - 1], cu_i]$
      **end if**
    **else**
      ▷ Erase False-Negative Type (a)
      **if** $d(cu_{i-1}, cu_i) \leq d_{min}$ **then**
        $cu_i \leftarrow \text{vereinige}(cu_i, cu_{i-1})$
        $CU_{smoothed} \leftarrow [CU_{smoothed}[0 : end - 1], cu_i]$
      **else**
        ▷ Don't accept $cu_i$. Erases False-Positives (b)
      **end if**
    **end if**
  **end for**
  **return** $CU_{smoothed}$
**end function**

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.

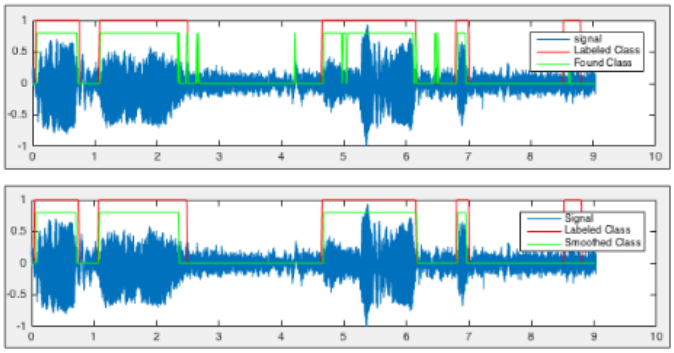


Abbildung 4.8: Klassifikation vor und nach dem Decision Smoothing

# 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.

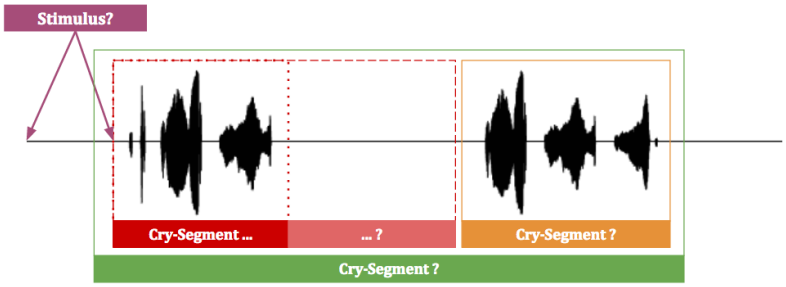


Abbildung 5.1: Mögliche Segmentierungen eines Signals

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[41, 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 4 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 3 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 4** 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.

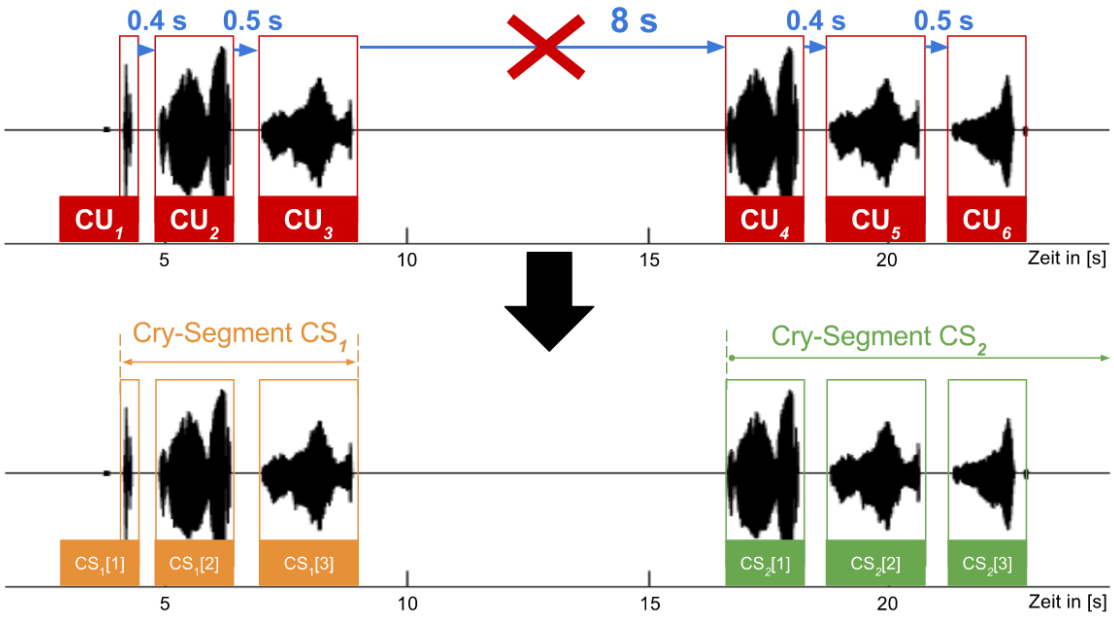


Abbildung 5.2: Ergebnis der Segmentierung mit einem Grenzwert von $t_s = 6\,\mathrm{s}$

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 **??**). 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 4.0.4 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 [41] 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 [38], 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 4.0.2 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 & Peter Gessler Eva Cignac, Romano Mueller. Pain assessment in the neonate using the Bernese Pain Scale for Neonates. *Early Human Development*, 78(2):125–131, 2004. Online unter `http://www.sciencedirect.com/science/article/pii/S0378378204000337`.

[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.

[16] Hodgkinson. Neonatal Pain Assessment Tool, 2012. Online unter http://www.rch.org.au/uploadedFiles/Main/Content/rchcpg/hospital_clinical_guideline_index/PAT%20score%20update.pdf.

[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.

[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] Taddio Nulman. A revised measure of acute pain in infants. *J Pain Symptom Manage*, 10:456–463, 1995. Online unter http://geriatricphysio.yolasite.com/resources/Modified%20Behavioral%20Pain%20Scale%20(MBPS)%20in%20infants.pdf.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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/.

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

[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% |

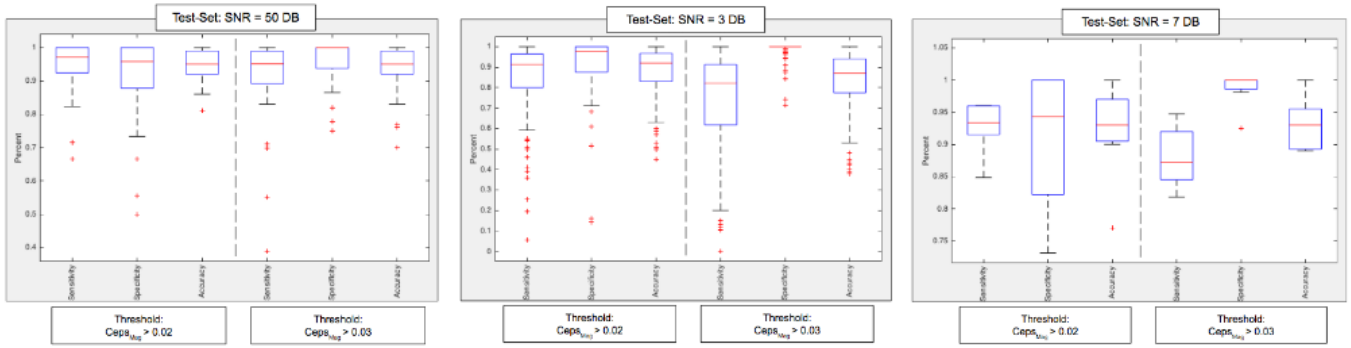


Abbildung .1: Boxplot-Auswertung über Sensitivity, Specificity und Accuracy der beiden VAD-Modelle