JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Echtzeit-Unterscheidung von Musik und Sprache in Online-Radiostreams

BACHELORARBEIT
(Projektpraktikum)

zur Erlangung des akademischen Grades

Bakkalaureus/Bakkalaurea der technischen Wissenschaften

im Bachelorstudium

INFORMATIK

Eingereicht von:
Georg Breitschopf, 0557147

Angefertigt am:
Institut für Computational Perception

Betreuung:
Univ.-Prof. Dr. Gerhard Widmer

Mitbetreuung:
Dipl.-Ing. Klaus Seyerlehner

Linz, Februar 2009

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Bakkalaureatsarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, Februar 2009 Georg Breitschopf

Kurzfassung

Die Unterscheidung von Sprache und Musik ist schon seit jeher ein wichtiges und interessantes Gebiet im Bereich der Informationsgewinnung aus Audio- und Videodaten. Wurden diese Aufgaben in der Vergangenheit noch von Menschen durchgeführt, so ist es aber aufgrund einer immer größer werdenden Menge an Multimediadaten mittlerweile unmöglich diese Daten manuell zu klassifizieren. Abhilfe sollen dabei Systeme zur automatischen Unterscheidung von Sprache und Musik schaffen, welche basierend auf dem Audiosignal und ohne die Verwendung von Metadaten diese Klassifizierung durchführen.

Im Rahmen dieser Arbeit wird ein Musik/Sprache - Klassifizierer zur automatischen Echtzeitanalyse von Online-Radiostreams präsentiert. Motiviert durch das implementierte Empfehlungssystem im Rahmen des FM4-Soundparkprojekts entstand die Idee, dieses auch für den FM4 Onlineradiostream zur Verfügung zu stellen. Um dieses aber sinnvoll einsetzen zu können, ist es aber zunächst notwendig und sinnvoll zu bestimmen, ob Musik gespielt wird oder nicht.

Nach einer kurzen Einführung in die Notwendigkeit der Musik/Sprache Unterscheidung werden die wesentlichen Grundlagen der Musik/Sprache Unterscheidung, welche ein typischer Anwendungsfall der Mustererkennung ist, erläutert. Ein Überblick über existierende Systeme soll das breite Spektrum an Umsetzungsmöglichkeiten aufzeigen und wichtige Informationen für den Vergleich der Systeme liefern.

Die durchgeführte Evaluierung des implementierten Systems ergab eine Klassifizierungsgenauigkeit von 90,14 %. Auf die Stärken und Schwächen des Systems sowie Verbesserungsmöglichkeiten wird in einer weiterführenden Analyse der Evaluierungsergebnisse eingegangen.

Abstract

The dicrimination of speech and music has always been an important and interesting area in the field of music information retrieval. These tasks has been done by humans in the past, but considering the growing amount of multimedia data, the task of manual classifying these data is impossible. Remedy will be given by systems that automatically discriminate between speech and music by just using the audio signal and without the use of meta data.

As a part of this thesis a music/speech discriminator for automatic real-time analysis of online radio streams will be presented. Motivated by the implemented recommendation system in the context of the FM4 Soundpark project the idea arise, to make this system available for the FM4 online radio stream. In order to be able to use this system meaningfully, it is first necessary and reasonable to determine whether music is played or not.

After a short introduction into the necessity of music/speech discrimination the basic concepts of music/speech discrimination, which is a typical application of pattern classification, are described. An overview of existing systems should point out the broad spectrum of implementation possibilities and should supply important informations for the comparison of the systems.

The accomplished evaluation of the implemented system resulted in a classification accuracy of 90,14 %. The strengths and weaknesses of the system as well as improvement opportunities are discussed in a further analysis of the evaluation results.

Inhaltsverzeichnis

Kapitel 1:

Einführung

Die Unterscheidung von Sprache und Musik ist schon seit jeher ein wichtiges und interessantes Gebiet im Bereich der Informationsgewinnung aus Audio- und Videodaten. Wurden diese Aufgaben in der Vergangenheit noch von Menschen durchgeführt, so ist es aber aufgrund einer immer größer werdenden Menge an Multimediadaten mittlerweile unmöglich diese Daten manuell zu klassifizieren. Abhilfe sollen dabei Systeme zur automatischen Unterscheidung von Sprache und Musik schaffen, welche basierend auf dem Audiosignal und ohne die Verwendung von Metadaten diese Klassifizierung durchführen. Dabei reichen die Datenquellen von Radio- und TV-Sendungen bis hin zur großen Anzahl an gespeicherten Audiodaten wie etwa Musik.

In der Literatur werden einige Anwendungsgebiete der Musik/Sprache Unterscheidung beschrieben. Dazu gehören unter anderem [Khan and Al-Khatib, 2006]:

- Automatische Spracherkennung (ASR):
  Da sich, beispielsweise in Radiosendungen, Musik und Sprache häufig abwechseln, ist es sinnvoll, die Daten zuvor in Musik und Sprache zu unterscheiden, um geeignete Eingabedaten für die automatische Spracherkennung zu erhalten, was wiederum zu einer Verringerung der Fehlerrate und Vermeidung unnötiger Berechnungen bei „Nicht-Sprach“-Segmenten beiträgt.

- Inhaltsbasierte Indizierung und Suche:
  Die Erkennung von Musik und Sprache ermöglicht die Einordnung von Audiosegmenten in verschiedene Kategorien, wie z.B. Musik, Sprache oder Stil-

le. Diese Kategorisierung vereinfacht die Suche nach Musik- oder Sprachsegmenten erheblich.

- Sprechererkennung:
  Die Erkennung von Sprache in Audiodaten ermöglicht die Anwendung von Sprechererkennungstechniken zur Identifizierung und Verfolgung von verschiedenen Personen.

- Verbesserung der Kodierung von Audiodaten:
  Die Klassifizierung von Audiodaten in Musik, Sprache oder Stille kann zur Reduzierung der Bitrate bei stillen Audiosegmenten eingesetzt werden und führt daher zur einer Verbesserung der Audiokodierung.

- Verbesserung von Komprimierungstechniken:
  Manche Komprimierungstechniken sind besser geeignet für Sprache, andere wiederum sind besser geeignet für Musik. Eine automatische Klassifizierung der Audiodaten ermöglicht das Anwenden der entsprechenden Komprimierungstechnik.

- Hörgeräte:
  Automatisches Anpassen von Hörgeräten an verschiedenste Geräuschsituationen (Musik, Sprache, Lärm, Stille, Wind, usw.) würde die Verwendung von Hörgeräten erheblich vereinfachen, da nicht mehr manuell der Modus gewechselt werden müsste.

Ein weiteres Anwendungsgebiet der Musik/Sprache Unterscheidung ist der Einsatz in Radiogeräten zur automatischen Erkennung von Werbung und Sprache, da viele Hörer mehr an Musik als an Werbung oder Gesprächen in Radioübertragungen interessiert sind [Saunders, 1996].

[Seyerlehner et al., 2007] beschreiben ein System zur automatischen Erkennung von Musik in TV-Produktionen um den Anteil von Musik berechnen zu können. Dieses System ermöglicht dem österreichischen Rundfunk (ORF) eine exaktere Berechnung der Lizenzgebühren, welche an die Behörden zu entrichten sind.

Abhängig vom Einsatzgebiet sind die Anforderungen an ein Musik/Sprache Unterscheidungssystem unterschiedlich. Beispielsweise ist es in einem Multimediasystem nicht wichtig die Daten in Echtzeit zu verarbeiten. Daher kann in solchen

Systemen die Verarbeitungszeit langsamer sein als in Systemen, wie z.B. Radioempfänger, die eine Analyse in Echtzeit erfordern. Hier ist eine geringe Latenzzeit wichtig. [Saad et al., 2002]

In weiterer Folge werden in dieser Arbeit einige Grundlagen der Musik/Sprache Unterscheidung erläutert (siehe Kapitel 2 auf Seite 5). In der Vergangenheit wurden bereits einige Arbeiten zu diesem Thema durchgeführt. Einen Überblick über diese Arbeiten soll Kapitel 3 auf Seite 9 schaffen.

1.1 Zielsetzung

Im Rahmen des FM4 Soundparkprojekts, welches in Zusammenarbeit mit dem österreichischen Forschungsinstituts für Artificial Intelligence [1], dem Institut für Computational Perception [2] und FM4 [3] umgesetzt wurde, wurde ein System zur automatischen Anzeige von Empfehlungen zum aktuell abgespielten Musikstück im FM4 Soundpark [4] implementiert. Dabei entstand die Idee diese Empfehlungen als zusätzliche Information im FM4-Liveradio [5] anzuzeigen. Da aber ein Radioprogramm nicht nur aus Musik (ca. 2/3; siehe Tabelle 5.1 auf Seite 24) besteht, sondern auch Werbung, Nachrichten, usw. enthält, entstand die Anforderung, die entsprechende Klasse (Musik, Nicht-Musik) für ein Segment bestimmter Länge zu ermitteln. Entsprechend dieser Klassifizierung sollen Empfehlungen angezeigt (Musik) oder keine Empfehlung ausgegeben werden (Nicht-Musik).

Folgende Anforderungen bzw. Vorgaben wurden hinsichtlich des zu implementierenden Klassifizierers ausgegeben:

- Plattform: Unix/Linux
- Programmiersprache: C++
- Echtzeitklassifizierung des FM4-Liveradio

[1] `http://www.ofai.at`
[2] `http://www.cp.jku.at`
[3] `http://fm4.orf.at`
[4] `http://fm4.orf.at/soundpark`
[5] `http://fm4.orf.at/static/stream/index.html`

- Bestimmung der Klasse für 15 Sekunden lange Blöcke
- Möglichst geringe Latenzzeit
- Zu verwendendes Feature: Continuous Frequency Activation (CFA) [Seyerlehner et al., 2007]

Details zur Implementierung werden in Kapitel 4 auf Seite 19 beschrieben. Die Ergebnisse der durchgeführten Evaluierung bzw. die daraus gewonnenen Erkenntnisse sind in Kapitel 5 auf Seite 23 aufbereitet.

Kapitel 2:

Grundlagen der Musik/Sprache Unterscheidung

Zu entscheiden, ob es sich bei einem Audiosignal um Musik oder Sprache handelt, ist ein typisches Problem der Mustererkennung. In Anlehnung an [Bishop, 2006] wird der Prozess der Mustererkennung in mehrere Schritte unterteilt (siehe Abbildung 2.1 auf der nächsten Seite).

Um ein Audiosignal einer bestimmten Klasse zuordnen zu können, ist es zunächst notwendig, dieses in eine geeignete Form umzuwandeln. Zunächst wird das zu analysierende Audiosignal abgetastet und in ein entsprechendes digitales Audiosignal umgewandelt. Abhängig von den jeweiligen Anforderungen des Mustererkennungssystems werden unterschiedliche Abtastraten verwendet. Zumeist ist eine Abtastrate von 11025 Hz aber völlig ausreichend.

Es ist klar, dass die Bestimmung der Klasse nicht für das gesamte Audiosignal erfolgen kann, da sich einerseits das Audiosignal über mehrere Minuten bzw. Stunden erstrecken kann und sich andererseits die zu bestimmenden Klassen innerhalb des Signals häufig abwechseln. Daher werden die zu analysierenden Daten in kurze Segmente (*Frames*) unterteilt. Als Segmentlänge haben sich Werte im Bereich zwischen 10 ms und 50 ms als geeignet erwiesen, weil dadurch garantiert wird, dass das Audiosignal in diesem Bereich stationär ist.

Die Bestimmung der korrekten Klasse basierend auf dem reinen Audiosignal ist meist nicht möglich. Daher werden für jedes Segment bestimmte Eigenschaften (*Features*) aus dem Audiosignal extrahiert. Damit soll eine Vereinfachung des

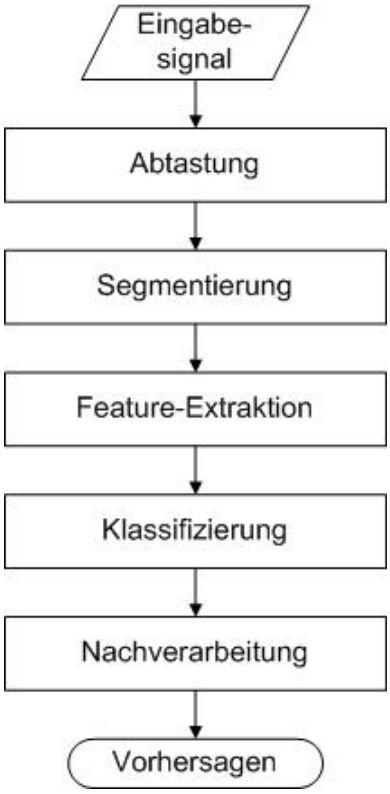

Abbildung 2.1: Schritte des Mustererkennungsprozess

Mustererkennungsproblems erreicht werden, was mit einer Reduzierung von Daten, Rechenzeit und Dimension der Daten einhergeht. Dabei ist allerdings zu beachten, dass es zu Informationsverlust und zu einer Vereinfachung des Problems kommen kann und dadurch die Genauigkeit des Systems beeinträchtigt wird. Generell lassen sich Features bzgl. ihrer Berechnungsgrundlage in zwei Bereiche unterteilen:

1. Features, welche aus dem Zeitbereich berechnet werden. Die Berechnungsgrundlage bilden dabei die Abtastwerte (Samples).

2. Features, welche aus dem Frequenzbereich berechnet werden. Als Berechnungsgrundlage dient das Spektrum des Signals, welches mit einer Fourier Transformation erzeugt wird.

Diese Features charakterisieren jeden einzelnen Frame und sollten je Klasse unterscheidbare Werte liefern. Um aber die Semantik eines Audiosignals zu verdeutlichen, ist es notwendig die Veränderung der Feature-Werte über die Zeit

zu verfolgen. [Khan and Al-Khatib, 2006] unterscheiden daher zwei Ebenen der Feature-Berechnung:

1. Framebasierte Features (*frame-level Features*)
   Diese Features liefern für kurze Ausschnitte (Frames) des Signals Informationen über deren mögliche Klasse.

2. Clipbasierte Features (*clip-level Features*)
   Ein Clip besteht dabei aus aufeinanderfolgenden Frames und hat meist eine Länge von mehreren Sekunden. Clipbasierte Features sollen nun Auskunft darüber geben, wie sich framebasierte Features über die Zeit verändern. Die Clipgrenzen ergeben sich entweder aus der Segmentierung, sodass der Inhalt eines Clips einer Klasse zuordenbar ist, oder werden auf eine fixe Größe festgesetzt.

Jedem Segment wird nun basierend auf den ermittelten Feature-Vektoren die entsprechende Klasse zugeordnet. Das Zuordnen der korrekten Klasse zu neuen Daten wird auch als Verallgemeinerung (*generalization*) bezeichnet. Eine Möglichkeit zur Durchführung der Klassifizierung ist der Einsatz von selbstdefinierten Regeln, welche aufgrund des vorhandenen Datenmaterials erstellt werden. Damit ist meist eine schnelle und einfache Klassifizierung möglich. Dabei kann es aber zu einem Regelwildwuchs kommen was wiederum zu schlechten Ergebnissen führen kann. Ein Verbesserung kann mit Algorithmen basierend auf maschinellen Lernen (Machine-Learning-Algorithmen) erreicht werden. Dabei werden zunächst in einer Lernphase (*learning phase*) mit Hilfe von Trainingsdaten (*training set*) die Parameter eines lernfähigen Modells bestimmt. Zu diesem Zweck werden die Feature-Vektoren mit den entsprechenden tatsächlichen Klassen annotiert und als Trainingsdaten verwendet. Diese Art des Lernens wird auch als supervised learning bezeichnet. Nachdem das Modell erstellt wurde, wird mit Hilfe von Evaluierungsdaten (*validation set*) das erzeugte Modell evaluiert. Oft passiert es, dass das gelernte Modell zu sehr an die Trainings- und Evaluierungsdaten angepasst ist. Dieses Problem wird als *over-fitting* bezeichnet. Um nun eine Aussage über die Qualität des Modells geben zu können, ist es notwendig dieses anhand von unabhängigen Testdaten (*test set*) zu überprüfen. Als Klassifizierer wird dann jenes Modell gewählt, welches die besten Ergebnisse liefert. Das Ergebnis des Klassifizieres sind dann die entsprechenden Vorhersagen (*predictions*) der jeweiligen Klassen.

Abhängig vom Kontext des Klassifizierungsproblems kann mittels Nachverarbeitung die Klassifizierungsgenauigkeit weiter gesteigert werden. Da die Segmentlänge meist nur wenige Millisekunden beträgt, ist es unwahrscheinlich, dass sich z.B. zwischen zwei längeren Musikblöcken ein Segment Nicht-Musik befindet und umgekehrt. Durch Anwendung von Glättungsfunktionen, welche aufgrund der Klassen der umliegenden Segmente die jeweilige Klasse des aktuell betrachteten Segments bestimmen, lassen sich solche Vorhersagefehler beseitigen. Diese Technik wird auch als *smoothing* bezeichnet. Durch Filterung von unplausibel kurzen Blöcken, die eine definierte Grenze unterschreiten, lassen sich weitere Klassifizierungsfehler beseitigen. Um solche Nachverarbeitungsschritte sinnvoll einsetzen zu können, ist eine Analyse der Audiodaten notwendig, welche die notwendigen, kontextabhängigen Information liefert.

Kapitel 3:

Verwandte Arbeiten

In der Vergangenheit wurden bereits eine Vielzahl an Arbeiten bzgl. Musik/-Sprache Unterscheidung erstellt. In diesem Kapitel werden einige dieser Arbeiten beschrieben und die dabei verwendeten Features und Klassifizierer angeführt. Ein Vergleich der einzelnen Ansätze untereinander ist kaum aussagekräftig, da unterschiedliche Testdaten verwendet wurden. Trotzdem soll dieses Kapital einen guten Überblick über die verschiedenen Umsetzungsmöglichkeiten und die dabei erzielten Genauigkeiten bieten.

Eine der ersten Systeme im Bereich der Musik/Sprache Unterscheidung wird in [Saunders, 1996] beschrieben. Das System wurde zur automatischen Unterscheidung von Musik und Sprache in FM Radioprogrammen entwickelt. Eine Analyse von Sprach- und Musiksignalen zeigte, dass sich die Wellenformen der beiden Signale stark unterscheidet. Es wurde festgestellt, dass der Energieunterschied zwischen Silben und Vokalen relativ groß ist, was als klare Struktur in der Wellenform eines Sprachsignals erkennbar ist. Auf Grund dieser Erkenntnisse wurde die zero-crossing rate (ZCR) zur Unterscheidung verwendet, welche direkt aus dem Zeitbereich eines Signals ermittelt werden kann. Mit Hilfe der ZCR lässt sich nach dem Prinzip der dominanten Frequenz auch die Frequenz mit der meisten Energie ermitteln. Weitere Analysen zeigten für Sprache deutliche Sprünge in der Wellenform der ZCR. Die Klassifizierung des Audiosignals wurde für Blöcke von 2,6 Sekunden Länge durchgeführt, wobei jeder Block aus 150 nicht überlappenden 16 ms Segmenten besteht. Für jedes dieser Segmente wurde die ZCR bestimmt. Als clip-level Features wurden die Standardabweichung, das dritte zentrale Moment des Mittelwerts, die Anzahl zero-crossings die einen bestimmten Schwellwert überschreiten sowie die Anzahl der zero-crossings über und unter

dem Mittelwert berechnet. Als Klassifizierer wurde ein multivariater Gaussian Classifier verwendet. Mit diesem Setup wurde eine Klassifizierungsgenauigkeit von 90 % erreicht. Durch Hinzufügen eines Features basierend auf der Energiekontur konnte die Leistung auf 98 % erhöht werden. Da alle Features aus dem Zeitbereich ermittelt werden, ist die Anwendung einer FFT nicht notwendig und gewährleistet dadurch eine relativ geringe Berechnungszeit. Das ist vor allem bei der Echtzeitanalyse von Audiostreams notwendig.

[Scheirer and Slaney, 1997] beschreiben in ihrer Arbeit die Konstruktion und Evaluierung eines Musik/Sprache Unterscheidungssystems. Es wurden 13 Features verwendet, von denen fünf die Varianzen von frame-level Features innerhalb eines ein Sekunden langen Fensters sind.

1. 4 Hz modulation energy
2. Percentage of „Low-Energy“ Frames
3. Spectral Rolloff Point
4. Varianz des Spectral Rolloff Point
5. Sepctral Controid
6. Varianz des Sepctral Controid
7. Spectral „Flux“ (Delta Spectrum Magnitude)
8. Varianz des Spectral „Flux“
9. Zero-Crossing Rate
10. Varianz der Zero-Crossing Rate
11. Cepstrum Resynthesis Residual Magnitude
12. Varianz der Cepstrum Resynthesis Residual Magnitude
13. Pulse Metric

Alle 13 Features wurden log-transformiert, um deren Verteilung an die einer Normalverteilung anzunähern. Zur Klassifizierung der Audiodaten wurden vier Klassifizierer getestet.

1. Multidimensional Gaussian maximum a posteriori (MAP) estimator
2. Gaussian mixture model (GMM)
3. Spatial partitioning scheme based on k-d trees
4. Nearest-neighbor classifier

Jedes der genannten Features wurde mittels 10-fold cross-validations getestet. Es wurden mehrere Kombinationen von Features sowie unterschiedliche Parametereinstellungen der Klassifizierer evaluiert. Das beste Ergebnis lieferten die besten drei Features (4 Hz modulation energy, Varianz des Spectral Flux und Pulse Metric) unter Verwendung des k-d spatial Klassifizierers mit einer Klassifizierungsgenauigkeit von 94,2 %. Um die Ergebnisse mit [Saunders, 1996] vergleichen zu können, wurde der Durchschnitt der Klassifizierungsergebnisse innerhalb von nicht überlappenden 2,6 Sekunden langen Fenstern berechnet. Durch diese Art der Nachverarbeitung konnten einige einzelne Klassifizierungsfehler beseitigt werden. Dadurch stieg die Klassifizierungsgenauigkeit auf 98,6 %.

In [Carey et al., 1999] wurden verschiedene Features miteinander verglichen. Analysen zeigten, dass auch Tonhöhe (Pitch) und Amplitude ein gutes Unterscheidungsmerkmal zwischen Musik und Sprache sind. Deshalb wurden einige Vergleichsexperimente mit folgenden Features durchgeführt:

1. Cepstral Coefficients
2. Delta Cepstral Coefficients
3. Amplitude
4. Delta Amplitude
5. Pitch

6. Delta Pitch

7. Zero-Crossing Rate

8. Delta Zero-Crossing Rate

Zur Feature-Berechnung wurde eine Fenstergröße von 10 ms verwendet. Die abgeleiteten Delta-Werte wurden mittels Abschätzung des Verlaufs der Feature-Werte über fünf aufeinanderfolgende Segmente ermittelt. Die Klassifizierung wurde mit Hilfe von Gaussian Mixture Models (GMM) durchgeführt. Sowohl Musik als auch Sprache wurden durch ein GMM repräsentiert, welches mit dem Expectation Maximisation (EM) Algorithmus trainiert wurde. Basierend auf den gleichen Daten wurden einige Experimente durchgeführt, wobei jeweils ein Feature mit seinem abgeleiteten Delta-Wert kombiniert wurde. Das beste Ergebnis erzielten Cepstral Coefficients und Delta Cepstral Coefficients mit einer equal error rate (EER) von 1,2 %. Obwohl des weit weniger komplexen Modells erzielten die Amplitudenwerte eine EER von 1,7 %. Pitch mit einer ERR von 4 % und Zero-Crossing mit einer ERR von 6 % konnten diese Genauigkeiten nicht erreichen.

[Williams and Ellis, 1999] beschreiben in ihrer Arbeit einen System basierend auf Posterior Probability Features. Ausgehend von drei Klassifizierern, welche auf neuralen Netzen basieren, werden für Fenster von 100 ms Länge die Wahrscheinlichkeiten von 54 Phonklassen ermittelt. Basierend auf dieser Folge von Posterior Phone Probabilities werden vier Features berechnet.

1. Mean per-frame entropy

2. Average probability „dynamism“

3. Background-label energy ration

4. Phone distribution match

Jedes dieser Features berechnet einen einzelnen Feature-Wert zur Bestimmung der Klasse innerhalb eines Blocks. Die Klassifizierung erfolgt mit Hilfe eines Gaussian Likelihood Ratio Tests. Dazu werden für alle Musik- und Sprachtrainingsdaten der Mittelwert und die Varianz getrennt berechnet. Für die Evaluierung wurden die Daten von [Scheirer and Slaney, 1997] verwendet. Es wurde jedes Feature

einzeln und Kombinationen dieser Features getestet. Für 15 Sekunden lange Audiofiles wurde eine Fehlerrate von annähernd 0 % erreicht. Um ihrer Ergebnisse mit denen von [Scheirer and Slaney, 1997] vergleichen zu können, wurde jede der Testdateien in sechs 2,5 Sekunden Blöcke unterteilt. Das beste Ergebnis lieferte die Kombination von Entropy, Dynamism und Energy mit einer Fehlerrate von 1,3 %.

Ein weiterer Ansatz zur Unterscheidung von Musik und Sprache wird in [El-Maleh et al., 2000] erläutert. Die Klassifizierung erfolgt dabei mittels eines schmalbandigen frame-level basierten Unterscheidungssystems. Als Features wurden Line Spectral Frequencies (LSF) verwendet, welche aus den Linear Prediction (LP) Koeffizienten berechnet werden. Durch weitere Kombination mit zero-crossing-basierten Features ergaben sich folgende vier Features:

1. Line Spectral Frequencies (LSF)

2. Differential Line Spectral Frequencies (DLSF)

3. Line Spectral Frequencies mit Higher Order Crossings (LSF-HOC)

4. Line Spectral Frequencies mit Zero Crossing Rate (LSF-ZCR)

Die Segmentlänge beträgt 20 ms. Aufgrund dieser Segmentlänge und der frame-level-basierten Klassifizierung ergibt sich eine Verzögerung von lediglich 20 ms. Dadurch ist das System auch für Echtzeitanwendungen geeignet. Zur Klassifizierung wurden zwei verschiedene Algorithmen verwendet.

1. Quadratic Gaussian Classifier (QGC)

2. k Nearest Neighbor Classifier (k-NN)

Unter Verwendung der oben genannten Features erzielte der k-NN Klassifizierer (k=3) die besten Ergebnisse. Für Musik wurde eine Klassifizierungsgenauigkeit von 79,2 %, für Sprache eine Klassifizierungsgenauigkeit von 82,5 % erreicht, was einer gesamten Klassifizierungsgenauigkeit von 80,85 % entspricht. Um die Genauigkeit des Systems weiter zu steigern, wurde zwei Arten der Nachverarbeitung getestet.

1. Fehlerkorrektur mit Verzögerung

2. Fehlerkorrektur ohne Verzögerung

Die Variante mit Verzögerung betrachtet jeweils das unmittelbar vorangegangene und das unmittelbar nachfolgende Segment des aktuellen Segments. Dazu ist eine Verzögerung von einem Segment notwendig. Die Variante ohne Verzögerung betrachtet die beiden vorangegangenen Segmente des aktuellen Segments. Um bei dieser Variante das Weiterreichen von Fehlern zu unterbinden, werden die entsprechenden zugewiesenen Klassen nach 15 Frames wieder zurückgesetzt. Bei beiden Varianten erfolgt die Entscheidung über die jeweilige Klasse basierend auf einer einfachen Mehrheitsregel (Modalwert). Experimente ergaben eine Verbesserung von 5 % bis 10 %, wenn die Klassifizierungsgenauigkeit bereits über 50 % betrug.

In [Jarina et al., 2001] wird ein System beschrieben, welches direkt mit den Daten des frequenzlimitierten MPEG-1 Bitstrom arbeitet um die aufwändige und komplexe Dekodierung des MPEG-1 Signals zu verhindern. Auf Basis der sogenannten Scalefactors, welche Informationen über das Signal für jedes Subband beinhalten, werden für 4 Sekunden lange Segmente die Einhüllende bzw. die Form des Audiosignals im Zeitbereich angenähert. Als Features wurden die Länge des größten Ausschlags bzw. die Anzahl der Ausschläge über einen gewissen Schwellwert verwendet. Zur Klassifizierung der Segmente wurde eine einfache Schwellwertmethode verwendet. Für die gesamten Testdaten konnte eine Klassifizierungsgenauigkeit von 90,9 % erreicht werden. Durch Eliminierung von einzelnen Musiksegmenten, welche sich zwischen zwei Sprachsegmenten befindet, konnte die Klassifizierungsgenauigkeit auf 92,93 % gesteigert werden. Eine genauere Betrachtung der Klassifizierungsergebnisse zeigte aber große Unterschiede bei Musikstücken mit einem starken Rhythmus. Um dieses Problem zu lösen, wurde in [Jarina et al., 2002] das Rhythm Metric Feature eingeführt, welches die Stärke des Rhythmus im Audiosignal quantifiziert. Durch Einführung dieses neuen Features konnte die Klassifizierungsgenauigkeit auf 95,81 % ohne Nachverarbeitung bzw. 97,71 % mit Nachverarbeitung gesteigert werden. Speziell bei Musikstücken mit viel Rhythmus konnte die Klassifizierungsgenauigkeit um ca. 11 % erhöht werden.

In [Karnebäck, 2001] wurde die Verwendung von Low Frequency Modulation Features zur Unterscheidung von Musik und Sprache untersucht. Als Features dienten 4 Hz Amplitude and standard deviation, 4 Hz Normalised Amplitude und 2-4

Hz Normalised Amplitude. Zur Klassifizierung wurden Gaussian Mixture Models (GMMs) verwendet. Für 2,5 Sekunden lange Blöcke wurde eine Klassifizierungsgenauigkeit von 93,6 % erreicht.

[Lu et al., 2001] beschreiben ein System zur Segmentierung eines Audiosignals in Musik, Sprache, Umgebungsgeräusch und Stille. Dazu wurden sechs Features verwendet.

1. High Zero-Crossing Rate Ratio (HZCRR)

2. Low Short-Time Energy Ratio (LSTER)

3. Spectrum Flux (SF)

4. LSP Distance

5. Band Periodicity (BP)

6. Noise Frame Ratio (NFR)

Die ersten drei Features basieren auf der Arbeit von [Scheirer and Slaney, 1997], die restlichen Features wurden neu eingeführt. Die Segmentierung bzw. Klassifizierung erfolgt in zwei Schritten. Zunächst wird basierend auf HZCRR, LSTER und SF mittels eines k-Nearest-Neigbor (KNN) Klassifizierer und einer Linear Spectral Pairs - Vector Qunatization (LSP-VQ) Analyse eine Unterscheidung in Sprache und Nicht-Sprache durchgeführt. Im zweiten Schritt werden all Nicht-Sprach-Segmente den Klassen Musik, Umgebungsgeräusch oder Stille zugeordnet. Dies erfolgt nach einem regelbasierten Schema. Zunächst wird anhand der Short-Time Energy und der Zero-Crossing Rate entschieden, ob es sich um Stille handelt. Für alle restlichen Segmente erfolgt die Unterscheidung in Musik und Umgebungsgeräusch mit Hilfe von BP, SF und NFR. Zur Klassifizierung wurde eine Segmentlänge von einer Sekunde gewählt. Um die Klassifizierungsgenauigkeit zu erhöhen, wurden mittels Smoothing unplausible Segmente entfernt. Für die Klassen Sprache, Musik und Umgebungsgeräusch wurde eine Genauigkeit von 96,51 % erreicht. Betrachtet man lediglich die Klassen Musik und Sprache wurden sogar 98,03 % korrekt klassifiziert.

In [Pinquier et al., 2002b] erfolgt die Klassifizierung in Musik und Sprache getrennt voneinander. Dies erfolgt durch zwei getrennte Klassifizierungssysteme welche Musik/Nicht-Musik bzw. Sprache/Nicht-Sprache mit Hilfe von GMMs unterscheiden. Zur Klassifizierung von Musik wurden Spectral Coefficients und für Sprache Mel Frequency Cepstral Coefficients (MFCC) verwendet. Diese Arbeit wurde ein [Pinquier et al., 2002a] weitergeführt und um die Features 4 Hz Modulation Energy, Entropy Modulation, Number of Segements und Segment Duration erweitert. Mit diesem System konnte eine Klassifizierungsgenauigkeit von 90,1 % erreicht werden.

[Saad et al., 2002] verwendeten Features, welche bereits in [Scheirer and Slaney, 1997] untersucht wurden.

1. Anteil von Low Energy Frames (LEF)

2. Spectral Roll-off Point (RO)

3. Spectral Flux (SF)

4. Zero Crossing Rate (ZCR)

5. Spectral Centroid (SC)

Die Klassifizierung erfolgt durch Vergleich der durchschnittlichen, relativen Verteilung der Features welche aus dem Verhältnis von prozentueller Verteilung jedes einzelnen Features und der maximalen Verteilung berechnet wird. Damit wurde eine Klassifizierungsgenauigkeit von 94,25 % erreicht.

[Wang et al., 2003] beschreiben ein einfaches System, welches nur ein Feature zur Klassifizierung verwendet. Sie verwenden das beschriebene Modiefied Low Energy Ratio (MLER) Feature. Die Klassifizierung erfolgt unter Verwendung eines Bayes MAP (Maximum A Posteriori) Klassifizierers. Des weiteren wird auch ein neues Modell zur Nachverarbeitung beschrieben, welches ähnlich einem Automaten arbeitet und kontextbezogen die Klasse bestimmt. Mit diesem System konnte eine Klassifizierungsgenauigkeit von 97 % für Musik und 98,4 % für Sprache erreicht werden. Zum Vergleich der kontextbezogenen Nachverarbeitung mit anderen Ansätzen, wurde jene von [El-Maleh et al., 2000] und [Jarina et al., 2002] ebenfalls implementiert. Damit konnte ein Klassifizierungsgenauigkeit von 94 % für Mu-

sik und 95,7 % für Sprache erreicht werden, was einen Unterschied von ca. 3 % bedeutet.

[Muñoz-Expósito et al., 2005] verwendet ebenfalls nur ein einziges Feature zur Unterscheidung von Musik und Sprache. Das neue Warped LPC-based Sectral Centroid (WLPC-SC) Feature basiert auf psychoakustischen Informationen und ist aufgrund der einfachen Berechnung und seiner Robustheit gut für Echtzeitklassifizierungssysteme geeignet. Die Klassifizierung erfolgt mittels GMMs welche mittels des EM-Algorithmus trainiert wurden. In der Arbeit wird von einer Klassifizierungsgenauigkeit von 93,2 % berichtet.

Eine weitere interessante Arbeit wird in [Pikrakis et al., 2006] beschrieben. Als Feature dient eine Variante der Spectral Entropy. Anstatt herkömmlicher Algorithmen zur Bestimmung der Klassen Musik und Sprache wird eine Technik aus der Bildverarbeitung verwendet, dass so genannte Region Growing. Zuerst werden bestimmte Segmente (Seeds) in definierten Abständen als Ausgangspunkt bestimmt. Iterativ werden dann diese Regionen vergrößert so lange die Standardabweichung der Spectral Entropy unterhalb eines bestimmten Schwellwertes liegen. Aufeinanderfolgende Segmente werden vereint und zu kurze Segmente eliminiert. Alle ermittelten Segmente werden der Klasse Musik, der Rest der Klasse Sprache zugeordnet. Für einen Schwellwert von 0,5, einer minimalen Segmentlänge von 3 Sekunden und eine Seed-Distanz von 2 Sekunden konnte eine durchschnittliche Klassifizierungsgenauigkeit von 93,38 % erreicht werden.

[Khan and Al-Khatib, 2006] haben versucht mit Hilfe von fuzzy C-Means Clustering die optimale Kombination aus folgenden Features zu finden:

- Anteil von Low Energy Frames (LEF)
- Spectral Flux (SF)
- Linear Predictive Coefficients (LPC)
- Range Of Zero-Crossings (R-ZC)
- Mittel (M-DWT) und Varianz (V-DWT) der Dicrete Wavelet Transformation

- RMS eines Lowpass Signals (RMS-LPS)
- Varianz der Mel Frequency Cepstral Coefficients (V-12MFCC)

Drei unterschiedliche Ansätze zur Klassifizierung der Evaluierungsdaten wurden getestet.

- Multilayer Perceptron (MLP)
- Radial Basis Functions (RBF)
- Hidden Markov Models (HMM)

Als Testdaten dienten sowohl Sprach- als auch Musikdaten aus unterschiedlichen Sprachen (Englisch, Urdu, Japanisch, Spanisch und Hebräisch). Das beste Ergebnis konnte MLP und den Features R-ZC, V-DWT, RMS-LPS, SF, LPC und V-4MFCC mit einer Klassifizierungsgenauigkeit von 100 % für die Sprache Englisch liefern.

[Seyerlehner et al., 2007] beschreiben einen neuen Ansatz zur Erkennung von Musik in Audiosignalen und speziell in TV Produktionen. Zu diesem Zweck wurde ein neues Feature entwickelt, dass sogenannte Continous Frequency Activation (CFA) Feature. Dieses neue Feature basiert auf der Tatsache, dass Musik mehr kontinuierlich aktive Frequenzen als Sprache enthält, welche gut als horizontale Linien im Spektrogramm erkennbar sind. Da dieses Feature einen einzigen Wert liefert, erfolgt die Klassifizierung mittels eines einfachen Schwellwertvergleichs. Basierend auf den nicht trivialen Testdaten konnte eine Klassifizierungsgenauigkeit von 89,93 % erreicht werden. Für die Daten von [Scheirer and Slaney, 1997] konnte eine Klassifizierungsgenauigkeit von 98,36 % erreicht werden. Des weiteren wurde auch das Potenzial der Nachverarbeitung gezeigt. Basierend auf einer Analyse der Testdaten wurde eine Smoothing-Logik implementiert, womit beachtliche Verbesserungen erzielt wurden.

Kapitel 4:

Implementierung

Wie bereits in Kapitel 1.1 auf Seite 3 angeführt, wurde der Musik/Sprache - Klassifizierer in C++ auf einer Unix/Linux - Plattform implementiert. Als IDE wurde Code::Blocks [6] verwendet.

4.1 Featureberechnung

Der Klassifizierer wurde als Single-Feature-Klassifizierer implementiert, d.h. der Klassifizierer bestimmt die jeweilige Klasse lediglich basierend auf einem einzigen Feature. Als Feature wurde das Continuous Frequency Activation (CFA) Feature [Seyerlehner et al., 2007] verwendet. Dieses wurde deshalb ausgewählt, da es sich hervorragend zur Bestimmung von Musik eignet. Das CFA Feature wird aus dem Spektrum des jeweilen Signals berechnet. Zur Umwandlung des Signals vom Zeitbereich in den Frequenzbereich ist eine Fouriertransformation notwendig. Diese wird mit Hilfe der unter der GNU General Public License (GNU GPL) [7] verfügbaren C-Funktionsbibliothek FFTW [8] („Fastest Fourier Transform in the West“) [Frigo and Johnson, 1998, Frigo and Johnson, 2005] durchgeführt.

[6] `http://www.codeblocks.org`
[7] `http://www.gnu.org/licenses/licenses.html`
[8] `http://www.fftw.org`

4.2 Verarbeitung des Onlinestreams

Um die Echtzeitklassifizierung des FM4-Onlinestreams [9] durchführen zu können, muss dieser in einer geeigneten Form (Blöcke vorgegebener Länge) aufgezeichnet werden. Zu diesem Zweck wird das unter der GNU GPL verfügbare Programm MPlayer [10] verwendet, da sich dieses auch relativ einfach als Kommandozeileprogramm verwenden lässt. Die ursprüngliche Idee, alle N Sekunden eine WAV-Datei zu erzeugen, musste allerdings aufgrund der fehlenden bzw. ungeeigneten Zeitinformation im FM4-Onlinestream verworfen werden. Um dieses Problem zu lösen, wurden statt dessen FIFOs (Namped Pipes), eine Technik aus der Unix-Interprozesskommunikation, verwendet. Im Unterschied zu einer Pipe wird diese allerdings nicht direkt zur Kommunikation zwischen Prozessen verwendet, sondern ist als Eintrag im Dateisystem verfügbar, auf den von mehreren Prozessen gleichzeitig zugegriffen werden kann. Die FIFO, welche beim Programmstart automatisch angelegt wird, wird dem MPlayer als Parameter für die Ausgabedatei übergeben. Der Stream wird mit der angegebenen Abtastrate (siehe Kapitel 4.4 auf der nächsten Seite) in ein mono 16-Bit PCM-Signal umgewandelt.

Listing 4.1: Beispielaufruf des MPlayer

```
mplayer -nolirc -really-quiet -vc null -vo null -af
    resample=11025:0:0,channels=1,format=s16ne -ao
    pcm:nowaveheader:file=FIFO mms://streamy.orf.at/fm4_live
```

Damit ist es nun ein leichtes die notwendigen Daten mittels eines einfachen Dateizugriffs im Klassifizierer einzulesen. Dabei wird die Einleseoperation so lange blockiert, bis die entsprechende Menge an Daten in der FIFO vorhanden ist. Aufgrund dieser Eigenschaft ist auch die Anforderung der Echtzeitanalyse erfüllt, da die Zeitdauer für die Berechnung des Featurewertes und die Bestimmung der Klasse wesentlich kürzer ist, als die Länge bzw. Dauer eines zu verarbeitenden Blockes.

[9] Stream-URLs: `mms://streamy.orf.at/fm4_live` oder `mms://stream0.orf.at/fm4_live`
[10] `www.mplayerhq.hu`

4.3 Ermittlung der Klasse

Wie in [Seyerlehner et al., 2007] beschrieben, wird ein CFA-Featurewert für einen Block von ca. 2,6 Sekunden länge berechnet. Dies bedeutet bei einer Blocküberlappung von 50 Prozent, dass für eine Segmentlänge von 15 Sekunden elf Featurewerte berechnet werden. Zur Ermittlung der jeweiligen Klasse eines Segments wird eine einfache Regel angewendet. Zunächst wird jür das zu klassifizierende Segment der Median der 11 Featurewerte berechnet. Danach erfolgt die Klassifizierung nach folgender Regel:

$$class(median_i) = \begin{cases} 1 & \text{wenn } median_i \geq t \\ 0 & \text{wenn } median_i < t \end{cases}$$

Der Wertebereich $\{0, 1\}$ der Funktion $class(median_i)$ entspricht dabei den Klassen Nicht-Musik (0) bzw. Musk (1). Als geeigneter Schwellwert t wurde 0,62 ermittelt. Die Ermittlung der Klassifizierungsregel und des Schwellwertes wird in Kapitel 5 auf Seite 23 detailiert erläutert.

4.4 Programmaufruf

Der Programmaufruf erfolgt über die Kommandozeile. Der Klassifizierer kann auf zwei verschiedene Arten verwendet werden, wobei des 1. Argument die Art der Verarbeitung definiert:

1. stream: Echtzeitklassifizierung eines Onlinestreams

2. file: Klassifizierung einer WAV-Datei

Die beiden Optionen werden nachfolgend kurz erläutert.

4.4.1 Stream

Wird der Klassifizierer mit der Option „stream“ aufgerufen, erfolgt die Echtzeitklassifizierung des angegebenen Streams. Die vorhergesagten Klassen werden am

Bildschirm ausgegeben. Folgende Argumente sind an den Klassifizierer zu übergeben:

1. Option: stream
2. Eingabedatei: Verzeichnis und Dateiname der FIFO
3. Stream-URL: URL des zu verarbeitenden Streams
4. Abtastrate in Hz
5. Segmentlänge in Sekunden
6. Analysedauer in Minuten (0 für unendlich)

Listing 4.2: Beispielaufruf des Klassifizierers mit Option „stream“

```
MSClassifier stream /home/FIFO mms://streamy.orf.at/fm4_live
    11025 15 0
```

4.4.2 Datei

Wird der Klassifizierer mit der Option „file“ aufgerufen, erfolgt die Klassifizierung der angegebenen WAV-Datei. Die vorhergesagten Klassen werden im selben Verzeichnis wie die FIFO in Textdateien ausgegeben. Folgende Argumente sind an den Klassifizierer zu übergeben:

1. Option: file
2. Eingabedatei: Verzeichnis und Dateiname der WAV-Datei. Die zu klassifizierende Audiodatei muss dabei als mono 8-, 16- oder 32-Bit PCM-Signal vorliegen.
3. Segmentlänge in Sekunden

Listing 4.3: Beispielaufruf des Klassifizierers mit Option „file“

```
MSClassifier file /home/wav_file.wav 15
```

Kapitel 5:

Evaluierung

Wie bereits in Kapitel 4 auf Seite 19 erläutert, wird für jeweils 15 Sekunden lange Blöcke die Klasse bestimmt. Da der Klassifizierer einem Musikempfehlungssystem vorgeschaltet werden soll, wurde diese Blocklänge gewählt, um eine geeignete Menge an Daten für dieses System zu garantieren. Je Block werden elf CFA-Featurewerte berechnet.

Wie aber nun die Klasse bestimmen? Welcher ist der geeignete Schwellwert? Wie gut ist der gewählte Klassifizierer im Vergleich mit anderen Ansätzen und welche Rückschlüsse können bezüglich Qualität des Klassifizierers bei unterschiedlichen Inhalten gezogen werden? Antworten auf diese Fragen werden in den nachfolgenden Abschnitten gegeben.

5.1 Evaluierungs- und Testdaten

Zu Evaluierungs- und Testzwecken wurden insgesamt 480 Minuten (acht Dateien zu je 60 Minuten) FM4-Radio mit einer Abtastrate von 11025 Hz als mono 16-Bit PCM-Signal aufgezeichnet. Die Evaluierungsdaten umfassen fünf Audiodateien zu je 60 Minuten. Die Testdaten bestehen aus den restlichen drei Dateien zu je 60 Minuten. Der gesamte Datenbestand wurde mit den entsprechenden Klassen (Musik, Nicht-Musik) annotiert, wobei jeweils 15 Sekunden lange Blöcke betrachtet wurden. Da es durchaus vorkommt, dass sich innerhalb eines Blockes Musik und Nicht-Musik abwechseln, wurde zur Bestimmung der Klasse immer der dominierende Teil herangezogen. Dabei wurde die Klasse Musik aber nur dann vergeben,

wenn auch tatsächlich ein Musikstück gespielt wurde. Werbung, FM4-Jingles oder durch Sprecher überlagerte Musikstücke wurden als Nicht-Musik annotiert. Die wichtigsten Kennzahlen zu den Evaluierungs- und Testdaten sind in Tabelle 5.1 nochmals zusammengefasst.

Datei	**Länge**	**Musik**	**Nicht-Musik**
Eval-1	60 min	77,08 %	22,92 %
Eval-2	60 min	72,08 %	27,92 %
Eval-3	60 min	39,17 %	60,83 %
Eval-4	60 min	79,17 %	20,83 %
Eval-5	60 min	71,67 %	28,33 %
Eval gesamt	300 min	67,83 %	32,17 %
Test-1	60 min	77,08 %	22,92 %
Test-2	60 min	49,58 %	50,42 %
Test-3	60 min	81,67 %	18,33 %
Test gesamt	180 min	69,44 %	30,56 %
Gesamt	480 min	68,64 %	31,37 %

Tabelle 5.1: Aufteilung von Musik und Nicht-Musik in den Evaluierungs- und Testdaten

5.2 Bestimmung der Klassifizierungsart

Da die Bestimmung der jeweiligen Klasse eines Blocks auf einer einfachen Regel basieren sollte, kamen drei Möglichkeiten in Frage:

1. Berechnung des arithmetischen Mittewerts der CFA-Werte je Block und Vergleich mit einem Schwellwert

2. Bestimmung des Median der CFA-Werte je Block und Vergleich mit einem Schwellwert

3. Bestimmung der Klasse für jeden einzelnen CFA-Wert innerhalb eines Blocks mittels Schwellwertvergleich und Ermittlung des Modalwerts (häufigste Klasse innerhalb eines Blocks)

Da es sich allerdings um ein zwei Klassenproblem handelt und die Anzahl der CFA-Werte je Block ungerade ist, gehört der Median immer zur häufigeren Klasse, d.h. dass Median und Modalwert immer das gleiche Ergebnis liefern. Deshalb wurde in weiterer Folge der Modalwert vernachlässigt.

Um nun den arithmetischen Mittelwert und den Median miteinander vergleichen zu können, wurden für die Evaluierungsdaten für beide Möglichkeiten die Vorhersagen für Schwellwerte von 0,41 bis 0,90 ermittelt und je Schwellwert eine Konfusionsmatrix erstellt. Mit Hilfe der ermittelten Konfusionsmatrizen wurden folgende Kennzahlen je Schwellwert berechnet:

- Klassifizierungsgenauigkeit: Gibt an, wie viele Blöcke korrekt klassifiziert wurden.
- Precision je Klasse: Gibt an, wie viele der als Musik/Nicht-Musik klassifizierten Blöcke auch tatsächlich Musik/Nicht-Musik Blöcke sind.
- Recall je Klasse: Gibt an, wie viele Musik/Nicht-Musik Blöcke auch als solche klassifiziert wurden.

Um nun zu bestimmen, ob arithmetischer Mittelwert oder Median das bessere Ergebnis liefern, wurden beide Möglichkeiten anhand folgender Kennzahlen miteinander verglichen:

- Höchste Klassifizierungsgenauigkeit
- Klassifizierungsgenauigkeit bei besten Precision/Recall (P/R)-Verhältnis der Klasse Musik ($Precision_{Musik} \cong Recall_{Musik}$)
- Klassifizierungsgenauigkeit bei besten Recall/Recall (R/R)-Verhältnis der beiden Klassen ($Recall_{Musik} \cong Recall_{Nicht-Musik}$)

Lieferten mehrere Schwellwerte das gleiche Ergebnis, so wurden die Werte vom niedrigsten Schwellwert herangezogen. Das beste P/R bzw. R/R Verhältnis lässt sich auch als Schnittpunkt der beiden Kurven Precision und Recall der Klasse Musik (siehe Abbildung 5.1 auf Seite 27) bzw. Recall der Klasse Musik und Recall der Klasse Nicht-Musik (siehe Abbildung 5.2 auf Seite 28) interpretieren. Die

Ergebnisse für das arithmetische Mittel sind in Tabelle 5.2, für den Median in Tabelle 5.3 gelistet.

	Genauigkeit		**P/R-Verhältnis Musik**		**R/R-Verhältnis**	
Datei	**t**	**%**	**t**	**%**	**t**	**%**
Eval-1	0,59	91,25 %	0,68	89,58 %	0,76	87,50 %
Eval-2	0,61	94,17 %	0,61	94,17 %	0,66	92,92 %
Eval-3	0,62	96,25 %	0,71	95,00 %	0,69	95,83 %
Eval-4	0,58	92,08 %	0,58	92,08 %	0,65	91,67 %
Eval-5	0,59	90,42 %	0,67	85,00 %	0,73	83,75 %
Ø	0,60	92,83 %	0,65	91,17 %	0,70	90,33 %
Gesamt	0,59	92,50 %	0,64	91,58 %	0,70	89,58 %

Tabelle 5.2: Ergebnisse für Evaluierungsdaten bei Anwendung des arithmetischen Mittels

	Genauigkeit		**P/R-Verhältnis Musik**		**R/R-Verhältnis**	
Datei	**t**	**%**	**t**	**%**	**t**	**%**
Eval-1	0,59	94,17 %	0,64	91,67 %	0,75	88,33 %
Eval-2	0,59	94,17 %	0,59	94,17 %	0,65	92,92 %
Eval-3	0,62	97,50 %	0,66	97,08 %	0,65	96,67 %
Eval-4	0,60	93,75 %	0,54	92,92 %	0,60	93,75 %
Eval-5	0,57	91,25 %	0,62	88,33 %	0,72	84,17 %
Ø	0,59	94,17 %	0,61	92,83 %	0,67	91,17 %
Gesamt	0,60	93,75 %	0,62	93,25 %	0,65	91,92 %

Tabelle 5.3: Ergebnisse für Evaluierungsdaten bei Anwendung des Median

Der Vergleich der Werte zeigt, das beide Varianten eine annähernd gleiche Klassifizierungsgenauigkeit aufweisen. Ein genauere Betrachtung der Ergebnisse zeigte aber klare Vorteile für den Median (siehe auch Abbildung 5.3 auf Seite 29). Der Median erzielte bei 18 der 21 Vergleichswerte eine im Bereich von 0,83 % bis 3,33 % höhere Klassifizierungsgenauigkeit. In den restlichen drei Fällen liefern beide Varianten das gleiche Ergebnis. Das arithmetische Mittel konnte bei keinem Vergleichswert die besseren Ergebnisse liefern. Eine graphische Gegenüberstellung der Vergleichswerte in Form von Diagrammen ist in Abbildung 5.4 auf Seite 30, Abbildung 5.5 auf Seite 31 und Abbildung 5.6 auf Seite 32 zu sehen. Auf Grund

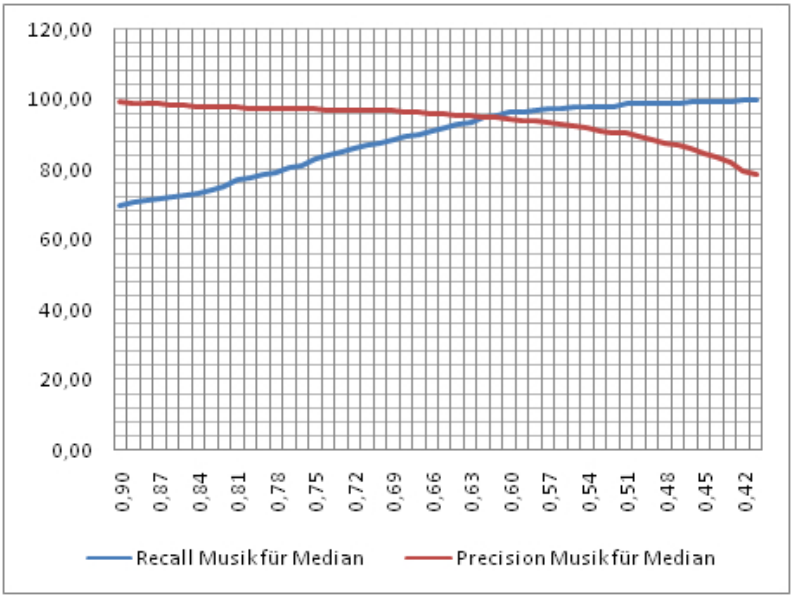

Abbildung 5.1: Precision/Recall-Plot der Klasse Musik für gesamtes Evaluierungsset bei Anwendung des Median

dieses Vergleichs und der Eigenschaft wesentlich robuster gegenüber Ausreißern zu sein, wurde der Median als entscheidender Wert für die Bestimmung der Klasse gewählt.

5.3 Bestimmung des Schwellwertes

Wie aus Tabelle 5.3 auf der vorherigen Seite leicht abzulesen, ist der Schwellwert bei der höchsten gefundenen Klassifizierungsgenauigkeit zumeist niedriger als beim besten P/R-Verhältnis und dieser wiederum zumeist niedriger als beim besten R/R-Verhältnis. Die ermittelten Schwellwerte beim besten P/R-Verhältnis und besten R/R-Verhältnis geben auch Auskunft über die Verteilung des absoluten bzw. relativen Fehlers. An dem Punkt (Schwellwert) an dem $Precision_{Musik} \cong Recall_{Musik}$ ist, ist der absolute Fehler annähernd gleich verteilt. An dem Punkt (Schwellwert) an dem $Recall_{Musik} \cong Recall_{Nicht-Musik}$ ist, ist der relative Fehler annähernd gleich verteilt (annähernd gleicher Fallout). Dies bedeutet, dass, je niedriger der Schwellwert gewählt wird, sich auch der relative Fehler für die Klasse Musik verringert bzw. für die Klasse Nicht-Musik erhöht.

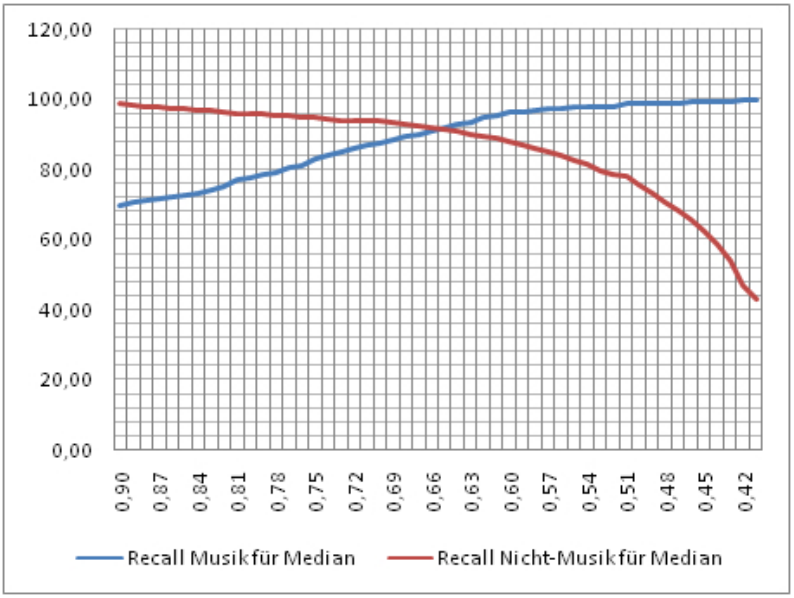

Abbildung 5.2: Recall/Recall-Plot für gesamtes Evaluierungsset bei Anwendung des Median

Um nun den idealen Schwellwert zu ermitteln wurden die Ergebnisse für den Median bzgl. Klassifizierungsgenauigkeit näher betrachtet. Dazu wurden die Werte für die Schwellwerte von 0,57 bis 0,68 nochmals näher betrachtet. Wie aus Tabelle 5.4 auf Seite 32 ersichtlich, ist der Unterschied bei den verschiedenen Schwellwerten marginal. Nun einfach jenen Schwellwert zu wählen, bei dem die höchste Klassifizierungsgenauigkeit ermittelt wurde, wäre nicht sinnvoll, da dadurch der Fehler bei Nicht-Musik zu hoch werden würde. Um eine bessere Verteilung des Fehlers zu erreichen, kamen jene Schwellwerte in Frage, die sich im Bereich zwischen gleichverteilten relativen und absoluten Fehler befinden. Da die Hauptaufgabe des Klassifizieres aber die Erkennung von Musik ist, erschien es vorteilhafter einen Schwellwert zu wählen, welcher einen niedrigeren relativen Fehler für die Klasse Musik gewährleistet. Daher wurde jener Schwellwert gewählt, der beim besten P/R-Verhältnis der Klasse Musik für den gesamten Evaluierungsdatenbestand ermittelt wurde. Dies ergab einen Schwellwert von 0,62.

5.4 Ergebnisse und Vergleich

Wie in den vorangegangenen Abschnitten ermittelt, erfolgt die Klassifizierung nun nach folgender einfachen Regel:

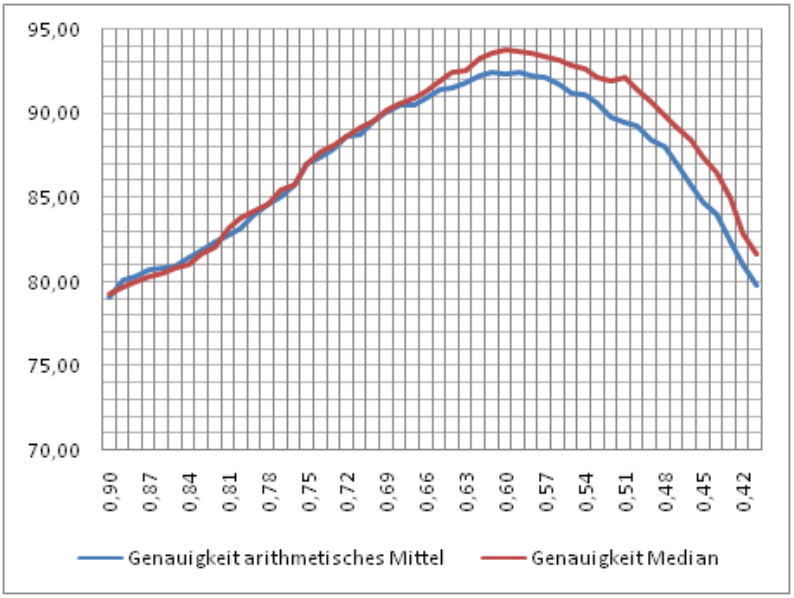

Abbildung 5.3: Klassifizierungsgenauigkeit bei Anwendung von arithmetischen Mittel und Median für gesamtes Evaluierungsset

$$class_i(median_i) = \begin{cases} Musik & \text{wenn } median_i \geq 0.62 \\ Nicht\_Musik & \text{wenn } median_i < 0.62 \end{cases}$$

Angewandt auf unabhängige Testdaten erzielte dieser einfache regelbasierte Klassifizierer (von nun an CFA-M genannt) beachtliche Ergebnisse (siehe Tabelle 5.5 auf Seite 33). Für die gesamten Testdaten wurde eine Klassifizierungsgenauigkeit von 90,14 % erreicht. Für die Testdatei Test-2 wurde sogar eine Klassifizierungsgenauigkeit von 94,58 % erzielt.

Um nun eine Aussage über die Qualität des CFA-M treffen zu können, war es notwendig, geeignete Vergleichswerte zu bestimmen. Zu diesem Zweck wurde WEKA [11] verwendet. Es wurden fünf unterschiedliche WEKA-Klassifizierer ausgewählt. Ein einfacher Nearest-Neighbor-Klassifizierer (IBk), MultilayerPerceptron (MLP) als Repräsentant für einen Klassifizierer basierend auf künstlichen neuralen Netzen, Support Vector Machines (SMO) aus der Gruppe der Kernel-Methoden, sowie REPTree und RandomForest aus der Familie der Entscheidungsbäume. Als Feature-Vektoren (je 15 Sekunden Block ein Vektor) wurden die elf CFA-Werte je Block verwendet. Für jeden dieser fünf Klassifizierer wurde die Klassifizierungs-

[11] `http://www.cs.waikato.ac.nz/ml/weka`

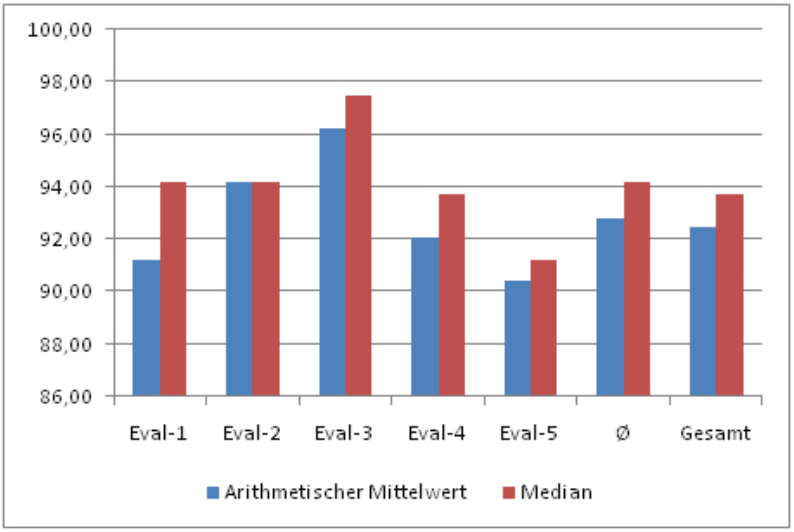

Abbildung 5.4: Ergebnisse für arithmetisches Mittel und Median bei höchster Klassifizierungsgenauigkeit

genauigkeit für die Testdaten berechnet. Als Trainingsdaten dienten die gesamten Evaluierungsdaten. Wie in Tabelle 5.5 auf Seite 33 bzw. Abbildung 5.7 auf Seite 34 ersichtlich, sind die Klassifizierungsgenauigkeiten der Machine-Learning-Algorithmen und des CFA-M kaum unterschiedlich. Das beste Gesamtergebnis erzielte SMO mit 91,53 %. Um weitere Vergleichsdaten zu gewinnen, wurde für jeden Klassifizierer 10-fold cross-validations auf Basis der Evaluierungs- und Testdaten durchgeführt. Aber auch hier sind kaum gravierende Unterschiede feststellbar (siehe Tabelle 5.6 auf Seite 33 bzw. Abbildung 5.8 auf Seite 34 und Tabelle 5.7 auf Seite 33 bzw. Abbildung 5.9 auf Seite 35).

Wie dieser Vergleich zeigt, ist der einfache, regelbasierte CFA-M mit der Leistung von Machine-Learning-Algorithmen vergleichbar. Grund dafür ist einerseits die Qualität des CFA-Features (siehe [Seyerlehner et al., 2007]), andererseits die Tatsache, dass beide Arten auf den selben Daten basierend die Vorhersagen treffen.

5.5 Interpretation der Evaluierungsergebnisse

Trotz einer respektablen Klassifizierungsgenauigkeit von 90,14 % war aber festzustellen, dass die Ergebnisse für die einzelnen Testdaten doch sehr unterschiedlich

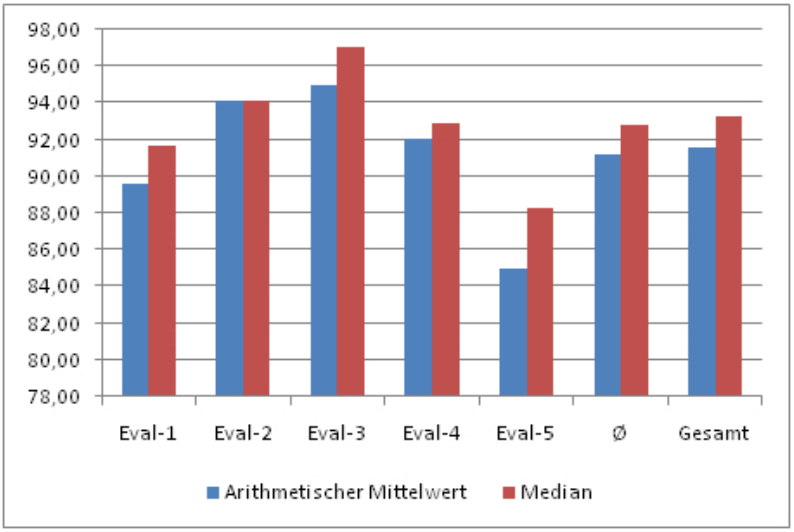

Abbildung 5.5: Ergebnisse für arithmetisches Mittel und Median bei besten P/R-Verhältnis

sind (86,67 % bei Test-1 bis 94,58 % bei Test-2). Da das CFA-Feature auf Basis kontinuierlich aktiver Frequenzen berechnet wird [Seyerlehner et al., 2007], liefert dieses abhängig vom jeweiligen Inhalt unterschiedliche Werte. Als problematisch erwiesen sich vor allem Musikgenres wie Rap, Hip-Hop und elektronische Musik. Da diese Musikrichtungen vor allem auf Beat basieren, enthalten diese wenige kontinuierlich aktive Frequenzen. Frequenzen, welche über die Zeit häufig aktiv sind, sind gut als horizontale Linien im Spektrogramm erkennbar. Wie gut in Abbildung 5.10 auf Seite 35 zu sehen ist, enthält Rock-Musik mehr kontinuierlich aktive Frequenzen als Hip-Hop (siehe Abbildung 5.11 auf Seite 36). Gleiches ist auch im Spektrogramm einer Stimme zu erkennen, wie in Abbildung 5.12 auf Seite 36 und Abbildung 5.13 auf Seite 37 gut zu sehen ist. Aufgrund dieser Eigenschaften liefert das CFA-Feature abhängig von der Musikrichtung unterschiedlich gut geeignete Werte für die Bestimmung der Klasse Musik. Auch Musikstücke, welche sich aus viel Gesang und weinig oder leiser, ruhiger Musik zusammensetzen, führen zu Klassifizierungsfehlern. Weitere Fehlerquellen sind Werbespots, da diese oft viel (Hintergrund-)Musik enthalten, sowie Moderationen mit Hintergrundmusik. Eine Gegenüberstellung der getroffenen Vorhersagen im Vergleich mit der tatsächlichen Klasse ist für die Testdaten in Abbildung 5.14 auf Seite 37, Abbildung 5.15 auf Seite 38 und Abbildung 5.16 auf Seite 38 visuell aufbereitet.

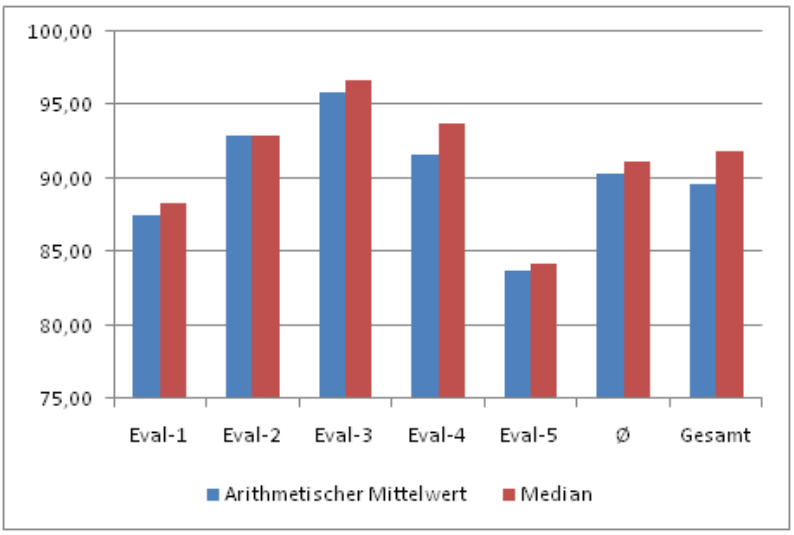

Abbildung 5.6: Ergebnisse für arithmetisches Mittel und Median bei besten R/R-Verhältnis

Datei	**0,68**	**0,67**	**0,66**	**0,65**	**0,64**	**0,63**
Eval-1	89,17 %	89,58 %	90,00 %	90,83 %	91,67 %	91,67 %
Eval-2	91,25 %	92,50 %	92,50 %	92,92 %	93,75 %	94,17 %
Eval-3	96,67 %	96,25 %	97,08 %	96,67 %	96,25 %	96,67 %
Eval-4	90,42 %	90,00 %	90,42 %	91,25 %	91,67 %	92,08 %
Eval-5	85,42 %	86,25 %	86,67 %	87,92 %	88,75 %	87,92 %
Gesamt	90,58 %	90,92 %	91,33 %	91,92 %	92,42 %	92,50 %
Datei	**0,62**	**0,61**	**0,60**	**0,59**	**0,58**	**0,57**
Eval-1	92,92 %	93,33 %	94,17 %	94,17 %	93,75 %	92,92 %
Eval-2	94,17 %	94,17 %	94,17 %	94,17 %	93,75 %	93,75 %
Eval-3	97,50 %	97,08 %	96,67 %	96,25 %	96,25 %	96,25 %
Eval-4	93,33 %	93,33 %	93,75 %	92,92 %	93,33 %	92,50 %
Eval-5	88,33 %	89,58 %	90,00 %	90,83 %	90,83 %	91,25 %
Gesamt	93,25 %	93,50 %	93,75 %	93,67 %	93,58 %	93,33 %

Tabelle 5.4: Klassifizierungsgenauigkeit bei Anwendung des Median auf Evaluierungsdaten für Schwellwerte von 0,57 - 0,68

Datei	**CFA-M**	**MLP**	**SMO**	**IBk**	**REP-Tree**	**Random-Forest**
Test-1	86,67 %	88,75 %	90,42 %	84,17 %	89,58 %	87,08 %
Test-2	94,58 %	91,67 %	91,67 %	89,58 %	91,67 %	92,50 %
Test-3	89,17 %	91,67 %	92,50 %	90,83 %	90,83 %	91,67 %
Gesamt	90,14 %	90,69 %	91,53 %	88,19 %	90,69 %	90,42 %

Tabelle 5.5: Klassifizierungsgenauigkeit bei Testdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (gelerntes Modell basierend auf gesamten Evaluierungsdaten)

Datei	**CFA-M**	**MLP**	**SMO**	**IBk**	**REP-Tree**	**Random-Forest**
Test-1	86,67 %	89,17 %	89,58 %	82,92 %	88,33 %	86,25 %
Test-2	94,58 %	93,75 %	95,00 %	90,42 %	92,92 %	93,75 %
Test-3	89,17 %	91,25 %	90,42 %	88,75 %	92,08 %	90,42 %
Gesamt	90,14 %	90,00 %	90,00 %	86,53 %	90,14 %	90,00 %

Tabelle 5.6: Klassifizierungsgenauigkeit bei Testdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (10-fold cross-validation)

Datei	**CFA-M**	**MLP**	**SMO**	**IBk**	**REP-Tree**	**Random-Forest**
Eval-1	92,92 %	92,50 %	93,75 %	92,92 %	91,67 %	94,17 %
Eval-2	94,17 %	91,25 %	94,17 %	91,25 %	91,25 %	92,50 %
Eval-3	97,50 %	97,08 %	96,67 %	95,00 %	96,67 %	96,25 %
Eval-4	93,33 %	94,58 %	94,17 %	92,50 %	93,75 %	95,83 %
Eval-5	88,33 %	90,83 %	90,00 %	87,92 %	88,33 %	87,08 %
Gesamt	93,25 %	94,08 %	94,08 %	92,17 %	93,50 %	93,00 %

Tabelle 5.7: Klassifizierungsgenauigkeit bei Evaluierungsdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (10-fold cross-validation)

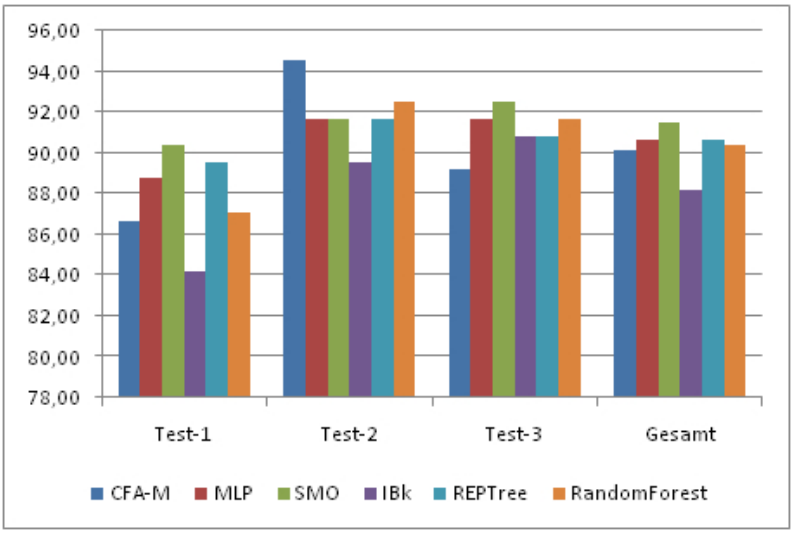

Abbildung 5.7: Klassifizierungsgenauigkeit bei Testdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (gelerntes Modell basierend auf gesamten Evaluierungsdaten)

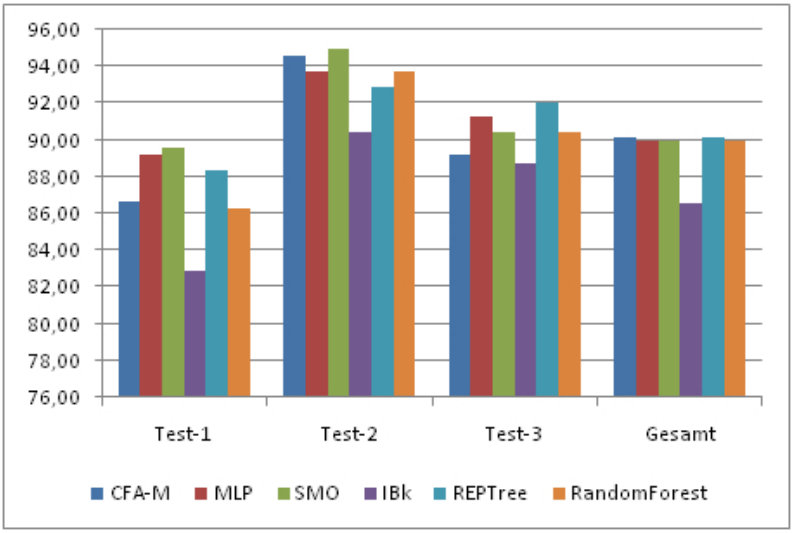

Abbildung 5.8: Klassifizierungsgenauigkeit bei Testdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (10-fold cross-validation)

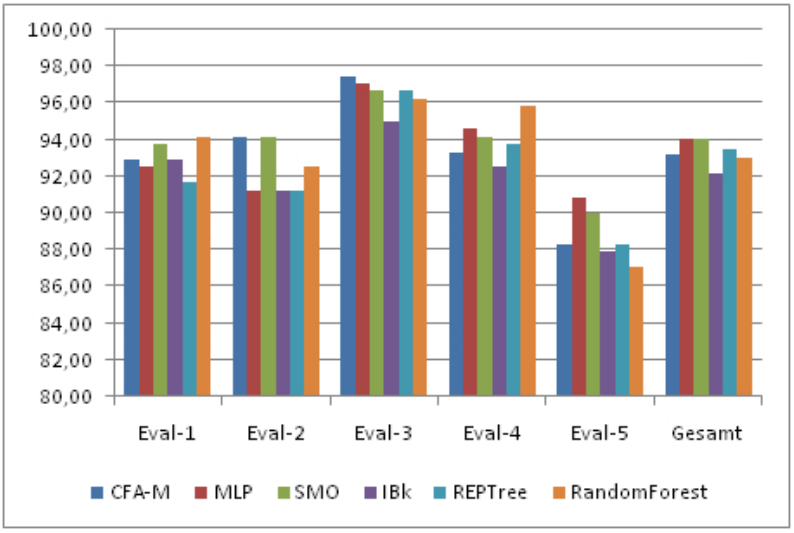

Abbildung 5.9: Klassifizierungsgenauigkeit bei Evaluierungsdaten für implementierten Klassifizierer und Machine-Learning-Algorithmen (10-fold cross-validation)

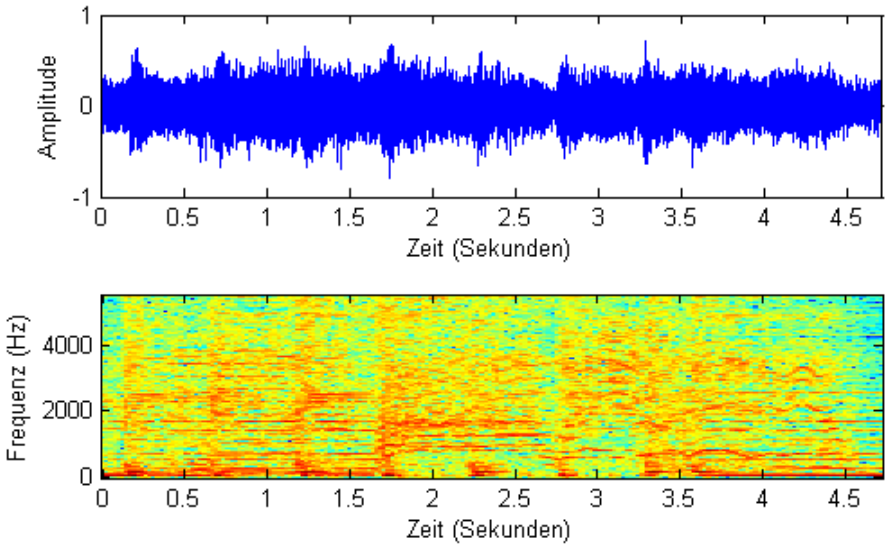

Abbildung 5.10: Wellenform und Spektrogramm eines ca. 4,7 Sekunden langen Audiosignals der Musikrichtung Rock

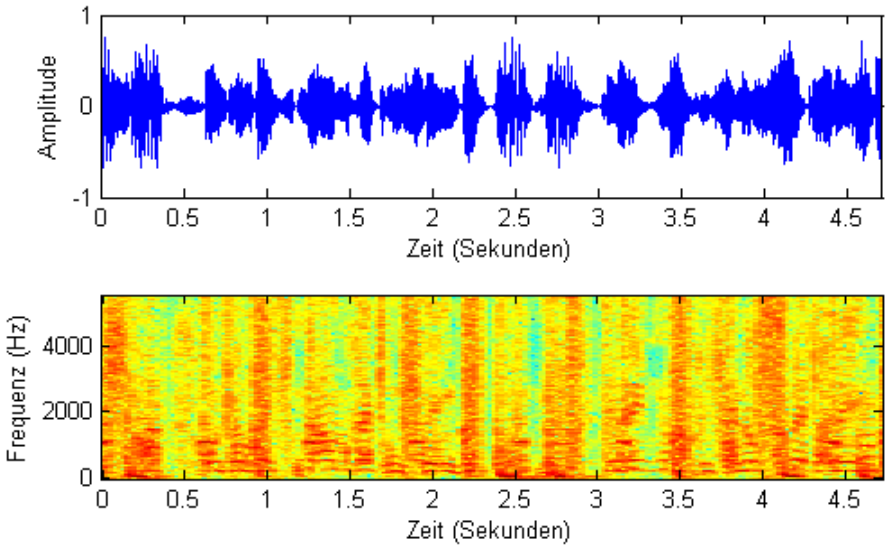

Abbildung 5.11: Wellenform und Spektrogramm eines ca. 4,7 Sekunden langen Audiosignals der Musikrichtung Hip-Hop

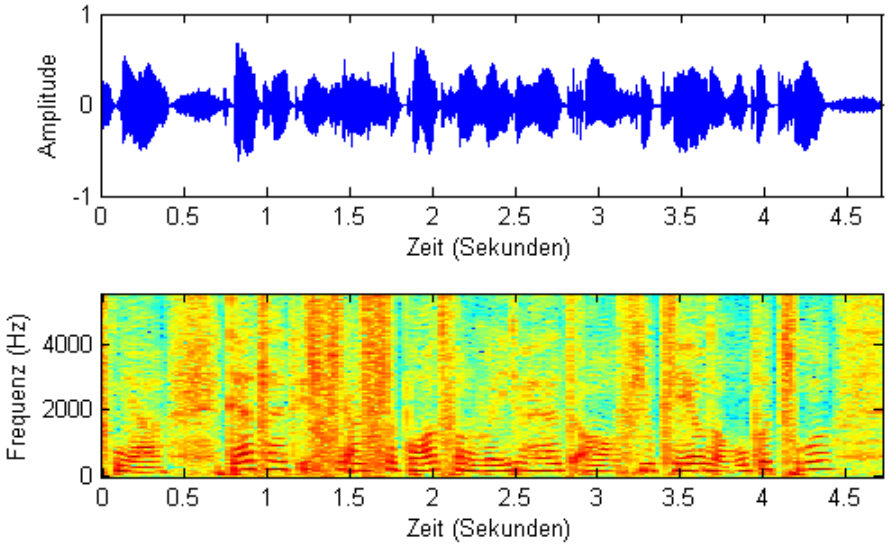

Abbildung 5.12: Wellenform und Spektrogramm eines ca. 4,7 Sekunden langen Audiosignals einer weiblichen Stimme

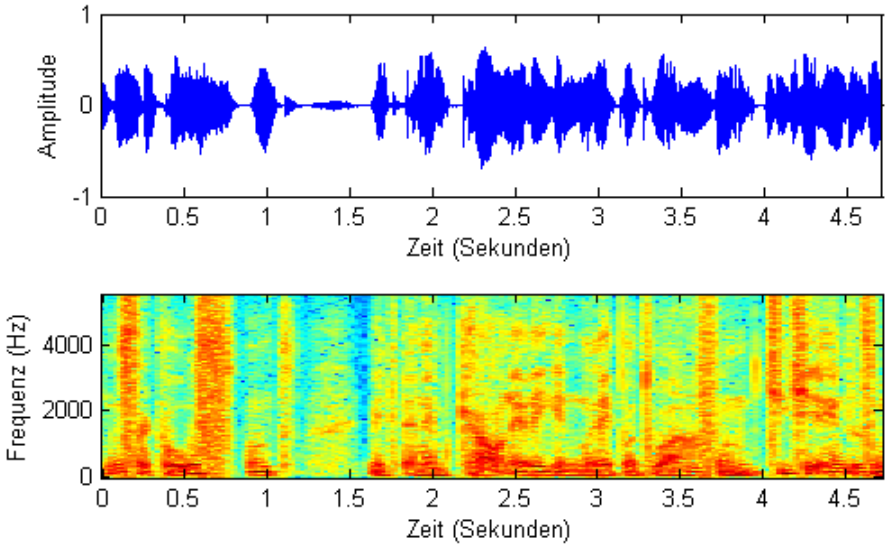

Abbildung 5.13: Wellenform und Spektrogramm eines ca. 4,7 Sekunden langen Audiosignals einer männlichen Stimme

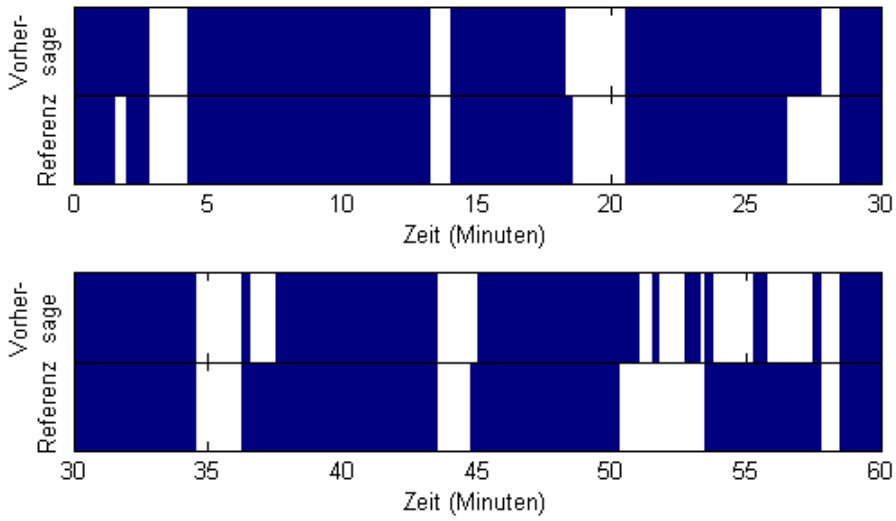

Abbildung 5.14: Visualisierung der Klassifizierungsergebnisse im Vergleich zu den tatsächlichen Klassen für die Testdatei Test-1. Dunkle Flächen kennzeichnen die Klasse Musik, helle Fläche die Klasse Nicht-Musik

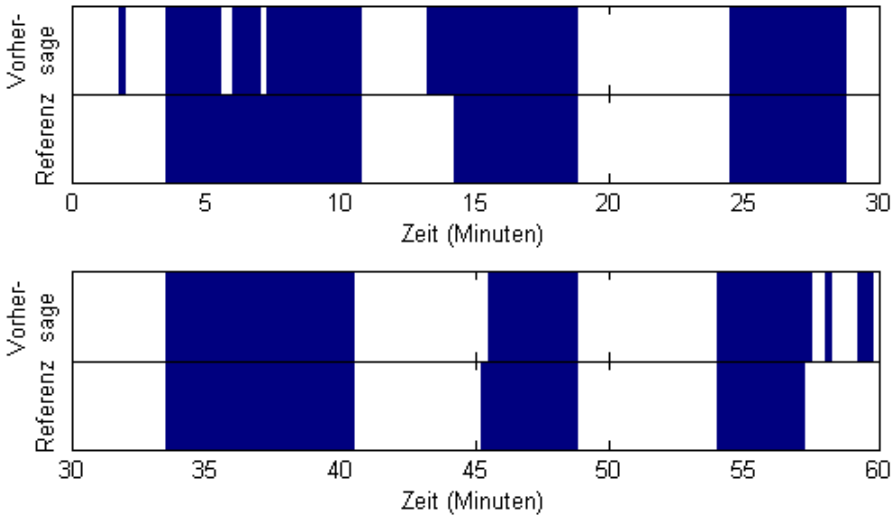

Abbildung 5.15: Visualisierung der Klassifizierungsergebnisse im Vergleich zu den tatsächlichen Klassen für die Testdatei Test-2. Dunkle Flächen kennzeichnen die Klasse Musik, helle Fläche die Klasse Nicht-Musik

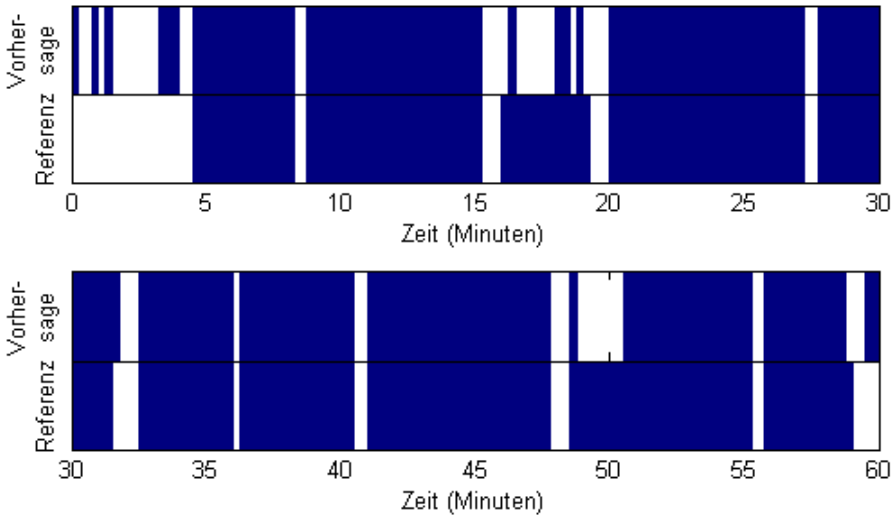

Abbildung 5.16: Visualisierung der Klassifizierungsergebnisse im Vergleich zu den tatsächlichen Klassen für die Testdatei Test-3. Dunkle Flächen kennzeichnen die Klasse Musik, helle Fläche die Klasse Nicht-Musik

Kapitel 6:

Zusammenfassung

Im Rahmen dieser Arbeit wurde ein Musik/Sprache - Klassifizierer zur automatischen Echtzeitanalyse von Online-Radiostreams präsentiert. Motiviert durch das implementierte Empfehlungssystem im Rahmen des FM4-Soundparkprojekts entstand die Idee dieses auch für den FM4 Onlineradiostream zur Verfügung zu stellen. Um dieses sinnvoll einsetzen zu können, ist es aber zunächst notwendig und sinnvoll zu bestimmen, ob Musik gespielt wird oder nicht.

In dieser Arbeit wurde auf die Grundlagen der Musik/Sprache Unterscheidung eingegangen, welche ein typischer Anwendungsfall der Mustererkennung ist. Einige zu diesem Thema verwandte Arbeiten wurden beschrieben. Eine detailierte Evaluierung des implementierten Klassifizierers ergab durchaus vergleichbare Klassifizierungsgenauigkeiten, obwohl die Klasse für jeweils 15 Sekunden lange Blöcke bestimmt wird. Auch der Vergleich mit diversen Machine-Learning-Algorithmen ergab ähnliche Ergebnisse.

Als Feature wird das Continous Frequency Activation (CFA) Feature verwendet, welches in [Seyerlehner et al., 2007] beschrieben wird. Die Klassifizierung jedes einzelnen Blocks erfolgt anhand eines einfachen Schwellwertvergleichs. Als Vergleichswert wird der Median aller CFA-Werte je Block herangezogen.

Kapitel 7:

Ausblick

Trotz der respektablen Klassifizierungsgenauigkeit von 90,14 % ergab eine genauere Analyse der Vorhersagefehler, dass diese vor allem bei stark rhythmischer Musik mit viel Beat, wie z.B. bei Rap oder Hip-Hop, auftreten. Die Einführung zusätzlicher Features, welche dem Problem der Rhythmus- bzw. Beaterkennung gerecht werden, könnten die Klassifizierungsgenauigkeit weiter steigern. Einige Ansätze zu diesem Problem wurden bereits in [Scheirer and Slaney, 1997], [Carey et al., 1999] und [Jarina et al., 2002] beschrieben.

Durch weitere Optimierung der Parametereinstellungen bei der Berechnung des CFA Features und Hinzufügen weiterer Features würde die Genauigkeit des Systems weiter gesteigert werden können. Aufgrund der bereits relativ weiten Verbreitung von Mehrkernprozessoren könnte die Laufzeit durch Techniken aus dem Bereich der parallelen Datenverarbeitung ebenfalls deutlich verbessert werden.

Literaturverzeichnis

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Carey et al., 1999] Carey, M. J., Parris, E. S., and Lloyd-Thomas, H. (1999). A comparison of features for speech, music discrimination. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, pages 149–152, Washington, DC, USA. IEEE Computer Society.

[El-Maleh et al., 2000] El-Maleh, K., Klein, M., Petrucci, G., and Kabal, P. (2000). Speech/music discrimination for multimedia applications. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, pages 2445–2448, Washington, DC, USA. IEEE Computer Society.

[Frigo and Johnson, 1998] Frigo, M. and Johnson, S. G. (1998). FFTW: An Adaptive Software Architecture for the FFT. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, volume 3, pages 1381–1384.

[Frigo and Johnson, 2005] Frigo, M. and Johnson, S. G. (2005). The Design and Implementation of FFTW3. In *Proceedings of the IEEE*, volume 93, pages 216–231.

[Jarina et al., 2001] Jarina, R., Murphy, N., O'Connor, N., and Marlow, S. (2001). Speech-Music Discrimination from MPEG-1 Bitstream. *Advances in signal processing, robotics and communications*, pages 174–178.

[Jarina et al., 2002] Jarina, R., O'Connor, N., Marlow, S., and Murphy, N. (2002). Rhythm detection for speech-music discrimination in MPEG compressed domain. In *Proceedings of the 14th International Conference on Digital Signal Processing (DSP'02)*, pages 129–132, Hellas, Greece.

[Karnebäck, 2001] Karnebäck, S. (2001). Discrimination between speech and music based on a low frequency modulation feature. In *European Conference on Speech Communication and Technology*, pages 1891–1894, Allborg, Denmark.

[Khan and Al-Khatib, 2006] Khan, M. K. S. and Al-Khatib, W. G. (2006). Machine-learning based classification of speech and music. *Multimedia Systems*, 12(1):55–67.

[Lu et al., 2001] Lu, L., Jiang, H., and Zhang, H. (2001). A robust audio classification and segmentation method. In *Proceedings of the ninth ACM international conference on Multimedia (MULTIMEDIA '01)*, pages 203–211, New York, NY, USA. ACM.

[Muñoz-Expósito et al., 2005] Muñoz-Expósito, J., Garcia-Galán, S., Ruiz-Reyes, N., Vera-Candeas, P., and Rivas-Peña, F. (2005). Speech/Music Discrimination Using a Single Warped LPC-Based Feature. In *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR 2005)*, pages 614–617.

[Pikrakis et al., 2006] Pikrakis, A., Giannakopoulos, T., and Theodoridis, S. (2006). A computationally efficient speech/music discriminator for radio recordings. In *Proceedings of of the 7th International Conference on Music Information Retrieval (ISMIR'06)*, pages 107–110, Victoria, Canada.

[Pinquier et al., 2002a] Pinquier, J., Rouas, J.-L., and André-Obrecht, R. (2002a). Robust Speech / Music Classification in Audio Documents. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP 02)*.

[Pinquier et al., 2002b] Pinquier, J., Sénac, C., and André-Obrecht, R. (2002b). Speech and Music Classification in Audio Documents. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*.

[Saad et al., 2002] Saad, E. M., El-Adawy, M. I., Abu-El-Wafa, M. E., and Wahba, A. A. (2002). A Multifeature Speech/Music Discrimination System. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE '02)*, volume 2.

[Saunders, 1996] Saunders, J. (1996). Real-time discrimination of broadcast speech/music. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '96)*, pages 993–996, Washington, DC, USA. IEEE Computer Society.

[Scheirer and Slaney, 1997] Scheirer, E. and Slaney, M. (1997). Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, page 1331, Washington, DC, USA. IEEE Computer Society.

[Seyerlehner et al., 2007] Seyerlehner, K., Widmer, G., Pohle, T., and Schedl, M. (2007). Automatic Music Detection in Television Productions. In *Proceedings of the 5th Workshop on Adaptive Multimedia Retrieval (AMR'07)*, Paris, France.

[Wang et al., 2003] Wang, W. Q., Gao, W., and Ying, D. W. (2003). A Fast and Robust Speech/Music Discrimination Approach. In *Proceedings of the International Conference on Information, Communications and Signal Processing.*

[Williams and Ellis, 1999] Williams, G. and Ellis, D. P. (1999). Speech/music discrimination based on posterior probability features. In *Proceedings of the 6th European Conference on Speech, Communication and Technology (EUROSPEECH'99)*, pages 687–690, Budapest, Hungary.