Klassifikation von Audio-Signalen

Diplomarbeit

von

THOMAS ROHDENBURG

Arbeitsbereich Nachrichtentechnik

Bremen, 27.05.03

Klassifikation von Audio-Signalen

Diplomarbeit

von

THOMAS ROHDENBURG

Ausgabetermin:	14.11.02
Abgabetermin:	27.05.03
Betreuer:	Dr.-Ing. Jörg Bitzer
Zuständiger Hochschullehrer:	Prof. Dr.-Ing. K.D. Kammeyer

Fachbereich Physik/Elektrotechnik (FB 1)
Arbeitsbereich Nachrichtentechnik
Postfach 33 04 40
D-28334 Bremen

Ich versichere, dass ich die Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Bremen, 27.05.03

. . . . . .

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Klassifikation von Audio-Signalen ist ein Forschungsbereich, der in den letzten Jahren sehr schnell vom Entwicklungsstadium in unser alltägliches Leben gelangt ist. Automatische sprachgesteuerte Informationsdienste sind heutzutage selbstverständlich, und auch am heimischen PC haben sich bereits komplexere Anwendungen, wie z.B. Diktierprogramme, etabliert. Derartige Systeme benötigen eine Spracherkennung und basieren daher auf der Zuordnung eines Eingangssignals zu einer Klasse (z.B. ein Wort, eine Silbe). Aufgrund der Zuordnung ist es anschließend möglich, den Programmablauf entsprechend zu beeinflussen und so eine Sprachsteuerung zu realisieren.
Ein anderes, intensiv untersuchtes Aufgabengebiet der Audio-Klassifikation ist die automatische Sprechererkennung. Hier werden auf der Grundlage vieler unterschiedlicher Sprachaufzeichnungen statistische Modelle für jeden Sprecher berechnet, mit denen eine spätere Identifikation eines Testsignals z.B. mit Hilfe einer Maximum-Likelihood Schätzung möglich ist. Die Anwendbarkeit der Sprechererkennung ist stark davon abhängig, wie sehr sich die Zuverlässigkeit für eine korrekte Klassifikation mit geeigneten Verfahren steigern lässt, damit z.B. auch sicherheitsrelevante Anwendungen wie Zugangsberechtigungs- oder Zeiterfassungssysteme realisierbar sind.
Weitere interessante Anwendungsmöglichkeiten ergeben sich aus der Klassifikation von Audio-Signalen in Musik-, Sprach- und Rauschsignale. Hiermit könnten z.B. große Audio-Archive, wie sie bei Rundfunk- und Fernsehanstalten vorhanden sind, analysiert und mit zusätzlichen Informationen über das Material (so genannten Meta-Daten) ausgestattet werden. Zahlreiche andere Anwendungsmöglichkeiten für den Einsatz der Signalklassifikation - auch außerhalb der Audio-Signalverarbeitung - sind denkbar und werden teilweise bereits in der Praxis realisiert.
In technischer Hinsicht muss für eine erfolgreiche Klassifikation gewährleistet sein, dass die statistischen Daten einer Signalklasse eine geringe Intra-Klassenvariation (Variation innerhalb einer Klasse) und eine hohe Inter-Klassenvariation (Unterschied zu den statistischen Daten einer anderen Klasse) aufweisen. Ist dies der Fall, so hat man eine hohe Diskriminationsfähigkeit zwischen den Klassen und steigert somit die Rate richtig klassifizierter Daten.
Als statistische Daten werden meistens spektrale Informationen aus dem Audio-Signal extrahiert und den Höreigenschaften des Menschen angepasst. Diese Daten bezeichnet man auch

als Merkmale (auch der englische Begriff „Features“ ist hierfür gebräuchlich). Bei den geläufigsten Merkmalen in der Sprechererkennung, den MFCC-Koeffizienten[1], sind die analysierten Spektraldaten auf die Mel-Skala[2] transformiert. Auch andere Verfahren der Merkmalextraktion, wie z.B. das im MPEG[3]-7 Standard definierte „Audio Spectrum Envelope“ (ASE)-Verfahren, ermöglichen eine Anpassung an das menschliche Hörempfinden und werden in dieser Arbeit für die Signalklassifikation verwendet.
Die Berechnung der Klassenmodelle erfolgt anhand iterativer Verfahren, von denen die drei bekanntesten,

- die Vektorquantisierung (VQ),
- das Hidden Markov Model (HMM) sowie das
- Gaussian Mixture Model (GMM)-Verfahren,

in der vorliegenden Arbeit untersucht werden.
Der Schwerpunkt liegt bei der Untersuchung des GMM-Verfahrens, das für die betrachteten Signalarten in Kombination mit Merkmalextraktionsverfahren aus dem MPEG-7 Standard die beste Diskrimination zwischen den Klassen ermöglichte.
Im nächsten Kapitel wird die Extraktion von Merkmalen aus den Audiodaten beschrieben. Anschließend folgt die theoretische Betrachtung der iterativen Modellberechnungs- und Klassifikationsverfahren. Im Kapitel 4 wird der Aufbau des Simulationssystems in Matlab sowie der Ablauf einer Klassenberechnung und Klassifikation behandelt. Auf die Auswertungen und Simulationen wird im fünften Kapitel eingegangen. Eine Zusammenfassung der Ergebnisse und ein Ausblick auf weitere Untersuchungen zur Klassifikation von Audio-Signalen erfolgt im Kapitel 6.

[1] MFCC = Mel Frequency Cepstral Coefficients
[2] Mel-Skala = Experimentell 1940 von Stevens und Volkmann ermittelte Skala zur subjektiven Wahrnehmung von Frequenzvervielfachungen durch den Menschen
[3] Motion Picture Expert Group

Kapitel 2

Merkmalextraktion

Das Ziel der Merkmalextraktion ist es, aus Audio-Signalen die wesentlichen Daten zu gewinnen, welche für die menschliche Unterscheidungsfähigkeit zwischen verschiedenen Signalen verantwortlich sind. Die zu analysierenden Audio-Signale müssen für die Berechnungsverfahren zunächst einmal in digital abgetasteter Form vorliegen. Die klanglichen Eigenschaften der Signale lassen sich am besten mit den Frequenzspektren kurzer, zeitlich begrenzter Abschnitte beschreiben. Aus diesen Spektren können mit unterschiedlichen Verfahren Daten extrahiert werden, welche die charakteristischen Eigenschaften der Audio-Signale möglichst gut in einer datenreduzierten Form repräsentieren. Derartige Daten bezeichnet man als Merkmale. In den folgenden Abschnitten werden unterschiedliche Verfahren der Merkmalextraktion beschrieben, unter anderem ein relativ neues Verfahren, mit dem eine stark datenreduzierte Darstellung eines Audio-Spektrogramms möglich ist.

2.1 Audio Spectrum Envelope

Das Merkmal „Audio Spectrum Envelope“ (ASE) ist im MPEG-7 Standard definiert und beschreibt das in Frequenzrichtung logarithmisch skalierte Leistungsdichtespektrogramm eines Audiosignals. Ein Spektrogramm ist die Frequenz-Intensität-Zeit-Darstellung eines Audiosignals (siehe Abb. 2.3), das durch das Betragsquadrat einer Kurzzeit-Fouriertransformierten erzeugt wird. Zur Extraktion des ASE müssen verschiedene Parameter festgelegt werden, wie z.B. untere/obere Bandgrenze (`loEdge`/`hiEdge`), Frequenzauflösung (`resolution`) und Zeitauflösung (entspricht in gewissen Grenzen `hopsize`). Die Schrittweite h wird mit dem Zeit-Parameter `hopsize` in Abhängigkeit von der Samplingfrequenz f_s des Audiosignals festgelegt:

$$\begin{aligned} f_s &= 16000\,\mathrm{kHz} \\ \texttt{hopsize} &= 0.01\,\mathrm{s} \\ \Rightarrow h &= 0.01 \cdot 16000 = 160\,\mathrm{Samples} \qquad \text{mit } h \in \mathbb{Z} \end{aligned} \tag{2.1}$$

Da h nur ganzzahlig sein kann, das Produkt aus Samplingfrequenz und `hopsize` aber rational ist, wird die Schrittweite h allgemein als ein Vektor definiert, dessen Mittelwert $\bar{h} = f_s \cdot$ `hopsize` ist.

$$\begin{aligned} f_s &= 22050\,\text{kHz} \\ \texttt{hopsize} &= 0.01\,\text{s} \\ \Rightarrow \bar{h} &= 0.01 \cdot 22050 \Rightarrow \mathbf{h} = [220, 221]\,\text{Samples} \end{aligned} \tag{2.2}$$

Durch das zyklische Abarbeiten des Schrittweitenvektors driftet die Analyse nicht vom gewünschten Analysezeitpunkt ab, so dass ein besserer Vergleich von Spektren - unabhängig von der Abtastrate des Audiosignals - möglich ist. Die Länge des Analysefensters l_w beträgt standardmäßig $3\cdot$`hopsize`, also z.B. $l_w = 30\,\text{ms} \cdot 16000\,\text{kHz} = 480\,\text{Samples}$ und wird mit einem Hamming-Fenster (Gl. B.4) gewichtet. Die Länge der FFT, n_{FFT} ist die nächsthöhere Zweierpotenz von l_w:
$l_w = 480\,\text{Samples} \Rightarrow n_{\text{FFT}} = 512\,\text{Samples}$.
Der mit einem Hamming-Fenster gewichtete Signalblock der Länge l_w wird auf die Länge n_{FFT} vergrößert und außerhalb des Fensters liegende Samples werden auf Null gesetzt (zero padding). Mit diesen Parametern wird dann eine Kurzzeit-Fouriertransformation (STFT, Short-Time Fourier Transform) durchgeführt.

$$\Phi_{\text{lin}}(n,t) = |\text{STFT}\{s\}|^2 = |\text{FFT}\left\{s(k,t)\right\}|^2 \quad \text{mit} \quad \begin{array}{rcl} t & : & \text{Blockindex} \\ k & : & \text{Zeitindex im Signalblock } s(t) \\ n & : & \text{Frequenzindex} \end{array} \tag{2.3}$$

Das mit Hilfe der Kurzzeit-Fouriertransformation erzeugte Spektrogramm $\Phi_{\text{lin}}(n,t)$ hat nun noch lineare Bandabstände. Da die Tonhöhenempfindlichkeit des menschlichen Gehörs eher einen logarithmischen Verlauf hat, bei der niedrige Frequenzen besser aufgelöst werden als hohe, wird die Frequenzachse des ASE logarithmisch skaliert. Hierzu müssen die (linearen) FFT-Koeffizienten entsprechend transformiert werden. Die logarithmischen Bandabstände werden durch die Gleichung

$$\text{edges}_{\text{log}} = 2^{rm} \cdot 1\,\text{kHz} \qquad \text{mit} \quad \begin{array}{c} \frac{m_l}{r} \leq m \leq \frac{m_h}{r} \qquad m \in \mathbb{Z} \\ r = \texttt{resolution} \end{array} \tag{2.4}$$

$$m_l = \log_2\left(\frac{\texttt{loEdge}}{1000\,\text{Hz}}\right) \qquad m_h = \log_2\left(\frac{\texttt{hiEdge}}{1000\,\text{Hz}}\right) \tag{2.5}$$

innerhalb des mit `loEdge` (untere Bandgrenze) und `hiEdge` (obere Bandgrenze) festgelegten Bereichs berechnet, wobei der Parameter `resolution` in Oktaven pro logarithmischen Band angegeben wird. Hat `resolution` den Wert 1, wird also innerhalb des Bereichs zwischen `loEdge` und `hiEdge` pro Oktave *ein* logarithmischer Koeffizient bestimmt; bei `resolution` = 1/8 dementsprechend 8 Koeffizienten pro Oktave. Hinzu kommen jeweils ein Koeffizient für die Energie unterhalb der Bandgrenze `loEdge` und oberhalb von `hiEdge` (siehe Abb. 2.1). Eventuell müssen die Grenzen angepasst werden, wenn sie nicht mit den durch $\text{edges}_{\text{log}}$

berechneten Werten zusammenfallen. Außerdem muss mindestens ein linearer FFT-Koeffizient in die logarithmischen Bänder fallen. Weitere Details zu den empfohlenen Parametern finden sich in [ISO02] sowie in der Auswertung im Abschnitt 5.1.

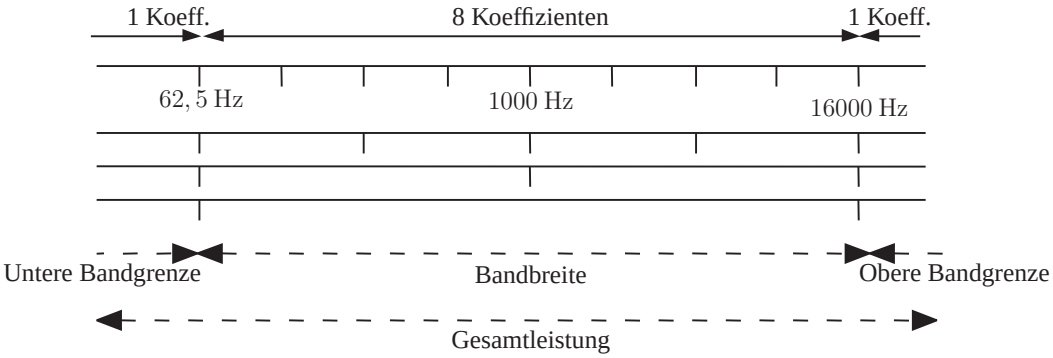


Abbildung 2.1: Audio Spectrum Envelope-Frequenzbänder

Ein FFT-Koeffizient des linearen Spektrums repräsentiert eine Bandbreite $D_f = f_s/n_{\mathrm{FFT}}$. Damit ergeben sich lineare Bandgrenzen bei

$$\mathrm{edges}_{\mathrm{lin}} = D_f \cdot n = \frac{f_s}{n_{\mathrm{FFT}}} \cdot n \qquad \text{mit } 1 \leq n \leq \frac{n_{\mathrm{FFT}}}{2}. \tag{2.6}$$

In einem logarithmischen Band werden nun mehrere der linearen FFT-Koeffizienten zusammengefasst[1]. Da sich nicht immer eine ganzzahlige Entsprechung der Bänder ergibt, werden die FFT-Koeffizienten, deren Abstand zur logarithmischen Bandgrenze kleiner als $D_f/2$ ist, durch einen linearen Faktor zwischen den benachbarten logarithmischen Bändern aufgeteilt (siehe Abb. 2.2).

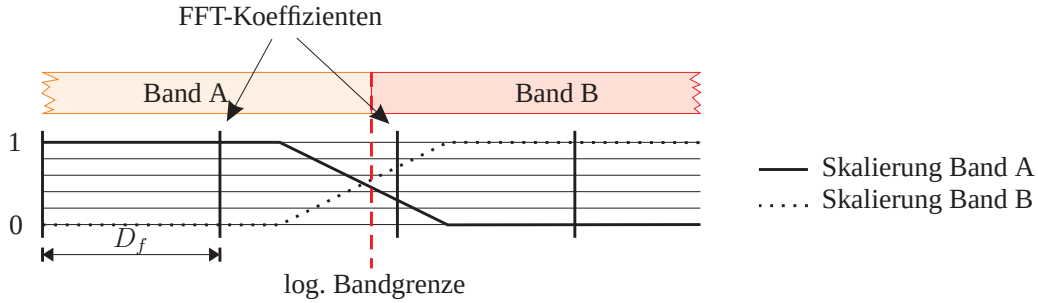


Abbildung 2.2: Gewichtungsmethode für Linear-zu-Logarithmus-Umwandlung

In der Praxis lässt sich das leicht mit Hilfe einer Transformationsmatrix **LT** bewerkstelligen, die mit dem linearen Spektrogramm multipliziert wird (siehe Gl. (2.8)). Die Transformationsmatrix

[1] Dieser Vorgang ist ähnlich zu der Gewichtungsmethode der Mel-Filterbank (siehe Abschnitt 2.3). Anstelle einer Dreieck-Gewichtung der Bänder wird hier jedoch eine trapezförmige Gewichtung der linearen FFT-Koeffizienten vorgenommen

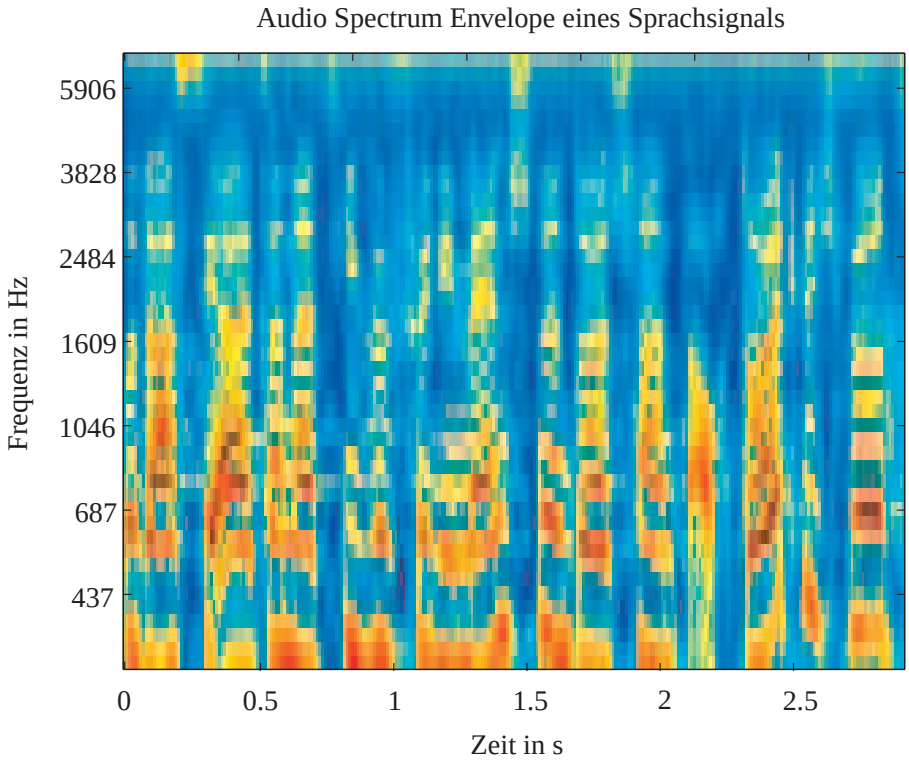


Abbildung 2.3: Spektrogramm eines Sprachsignals

enthält die Faktoren, mit der die linearen FFT-Koeffizienten in die logarithmischen Frequenzbänder einfließen. Die Spaltensumme von **LT** ist dabei 1.

$$\mathbf{LT} = \overbrace{\begin{pmatrix} 1 & 1 & 0.5 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0.5 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0.62 & 0.4 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0.38 & 0.6 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}}^{\text{Lineare FFT-Koeffizienten}} \quad \text{ASE-Koeffizienten} \tag{2.7}$$

Da **LT** nicht quadratisch ist, ist die Matrix nicht invertierbar und die Operation kann nicht rückgängig gemacht werden. Die Dimensionsreduktion von $n_{\mathrm{FFT}}/2$ auf N Frequenzbänder $(N < n_{\mathrm{FFT}}/2)$ ermöglicht eine erhebliche Einsparung an Rechenleistung für nachfolgende Verarbeitungsschritte (z.B. von 512 auf 39 Frequenzkoeffizienten). Dieser Vorteil überwiegt deutlich dem Informationsverlust, der durch die Zusammenfassung der höheren Frequenzbänder bewirkt wird.

$$\mathbf{\Phi_{log}} = \mathbf{LT} \cdot \mathbf{\Phi_{lin}} \qquad \text{mit} \quad \begin{aligned} \mathbf{LT} &\in \mathbb{R}^{N \times n_{\mathrm{FFT}}/2} \\ \mathbf{\Phi_{lin}} &\in \mathbb{R}^{n_{\mathrm{FFT}}/2 \times T} \\ \mathbf{\Phi_{log}} &\in \mathbb{R}^{N \times T} \end{aligned} \tag{2.8}$$

2.2 Dimensionsreduzierte Darstellung des ASE

In dem durch „Audio Spectrum Envelope“ erzeugten Spektrogramm sind noch viele Informationen enthalten, die für die Berechnung einer Klasse irrelevant sind und sich störend auswirken können. Sind die Trainingssignale einer Klasse beispielsweise in unterschiedlichen Umgebungen aufgenommen, so gibt es Beeinträchtigungen durch Hintergrundgeräusche, die nicht zur Signalklasse gehören und so zu einer Vergrößerung der Intra-Klassenvariation[2] führen.
Hinzu kommt, dass die Vektoren der spektralen Leistungsdichte eines jeweiligen Frequenzbandes über die Zeit häufig eine statistische Abhängigkeit zueinander haben. Somit enthält das ASE-Spektrogramm Redundanz, die zu keiner Verbesserung der Berechnung des Klassenmodells führt.
Das Merkmal „Audio Spectrum Projection“ ist eine Weiterverarbeitung des Merkmals „Audio Spectrum Envelope“. Es ermöglicht eine starke Dimensionsreduktion des Spektrogramms unter Beibehaltung der wichtigsten spektralen Anteile, die im Hinblick auf die Klassifikation die Information über das Signal beinhalten. Um zu einer weniger redundanten Darstellung des Spektrogramms zu gelangen, wird mit Hilfe der im MPEG–7 Standard spezifizierten Funktion „Audio Spectrum Basis“ eine Transformationsmatrix $\mathbf{V}_{\mathrm{red}}$ berechnet. Durch Multiplikation des normalisierten Spektrogramms mit $\mathbf{V}_{\mathrm{red}}$ kann damit eine reduzierte Darstellung, die Audio Spectrum Projection, gewonnen werden.

2.2.1 Audio Spectrum Basis

Zur Berechnung der Transformationsmatrix $\mathbf{V}_{\mathrm{red}}$ sind folgende Schritte notwendig. Zunächst werden die Intensitätswerte des Spektrogramms logarithmiert. Damit wird eine Anpassung an das menschliche Lautstärkeempfinden vorgenommen.

$$\mathbf{\Psi} \;=\; 10 \cdot \log_{10}\left(\mathbf{\Phi_{log}}\right) \qquad \text{mit } \mathbf{\Psi} \in \mathbb{R}^{N \times T} \tag{2.9}$$

Die Einhüllende $\boldsymbol{\psi}_{\mathbf{env}(1\times T)}$ wird bestimmt, indem für jeden Zeitrahmen des Spektrogramms $\mathbf{\Psi_t}_{(N\times 1)}$ die euklidische Norm (L2-Norm) berechnet

$$\psi_t = \text{L2-Norm}\{\mathbf{\Psi_t}\} \;=\; ||\mathbf{\Psi_t}||_2 = \sqrt{\sum_{n=1}^{N} |\Psi_t(n)|^2} \quad t = 1 \ldots T. \tag{2.10}$$

und dann zu einem Vektor

$$\boldsymbol{\psi}_{\mathbf{env}} \;=\; (\psi_1, \psi_2, \ldots, \psi_T) \tag{2.11}$$

zusammenfasst wird. Anschließend wird das Spektrogramm durch seine Einhüllende (euklidische Norm) geteilt.

$$\widetilde{\mathbf{\Psi}} \;=\; \left(\widetilde{\mathbf{\Psi}}_{\mathbf{1}}, \widetilde{\mathbf{\Psi}}_{\mathbf{2}}, \ldots, \widetilde{\mathbf{\Psi}}_{\mathbf{T}}\right) \qquad \text{mit } \widetilde{\mathbf{\Psi}}_{\mathbf{t}} = \frac{\mathbf{\Psi_t}}{\psi_t} = \frac{\mathbf{\Psi_t}}{\boldsymbol{\psi}_{\mathbf{env}}(t)}, \quad t = 1 \ldots T \tag{2.12}$$

[2] Variation der Trainingsdaten innerhalb einer Signalklasse

Nun wird das normierte und logarithmierte Spektrogramm $\widetilde{\mathbf{\Psi}}$ mit Hilfe der Singulärwertzerlegung (Singular Value Decomposition, SVD) in drei Matrizen zerlegt, so dass gilt

$$\widetilde{\mathbf{\Psi}}^{\top} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top}. \tag{2.13}$$

Nähere Angaben zur Berechnung und dem Aufbau der Matrizen $\mathbf{U}, \mathbf{S}$ und $\mathbf{V}$ folgen im Anhang A.1 (SVD).

Von den drei Matrizen ist für die Audio Spectrum Basis nur die Matrix $\mathbf{V}_{(N\times N)} \in \mathbb{R}^{N\times N}$ von Bedeutung. $\mathbf{V}$ ist eine orthonormale Matrix aus Basisvektoren, die den so genannten Parameterraum aufspannen [RJ93, Vac90]. Im Parameterraum lassen sich alle T Datenwerte des Spektrogramms durch Linearkombination der Basisvektoren darstellen. Die Basisvektoren sind die normierten Eigenvektoren des Matrizenprodukts

$$\widetilde{\mathbf{\Psi}}\widetilde{\mathbf{\Psi}}^{\top} \in \mathbb{R}^{N\times N} \tag{2.14}$$

Die (Basis-)Eigenvektoren in $\mathbf{V}$ sind (aufgrund einer Konvention bei der SVD-Zerlegung) in absteigender Reihenfolge nach der Größe der zugehörigen Eigenwerte sortiert. Verwendet man nun eine Untermenge von $K < N$ Eigenvektoren, so trifft man eine Auswahl der wichtigsten Komponenten des Spektrogramms.

$$\mathbf{V}_{\text{red}} \in \mathbb{R}^{N\times K} \qquad \text{mit} \quad \begin{array}{lcl} N & : & \text{Anzahl ASE-Spektralkoeffizienten} \\ K & : & \text{Anzahl Basisvektoren} \end{array} \tag{2.15}$$

Damit findet eine Dimensionsreduktion statt, die in Abb. 2.4 veranschaulicht ist: Die Eigenvektoren oder auch Basisvektoren von $\mathbf{V}$ sind zueinander orthogonal und spannen den Parameterraum auf. Stellt man nun fest, das die Ausdehnung des Raumes in eine Richtung besonders schwach ist, kann man diese Komponente in guter Näherung vernachlässigen. In dem abgebildeten Beispiel ergibt sich aus einem Raum also eine Ebene (der Unterraum $\mathbb{R}^{3\times 2}$) und damit die dimensionsreduzierte Darstellung des Originalraumes.
Die mit der Funktion Audio Spectrum Basis berechnete Matrix $\mathbf{V}_{\text{red}}$ ist damit das orthogonale Koordinatensystem des Spektrogramms, das auf die wichtigsten $K < N$ Komponenten reduziert wurde.

Geometrische Deutung der Dimensionsreduktion durch
Audio Spectrum Basis und Audio Spectrum Projection
mit Hilfe der Singulärwertzerlegung (SVD)

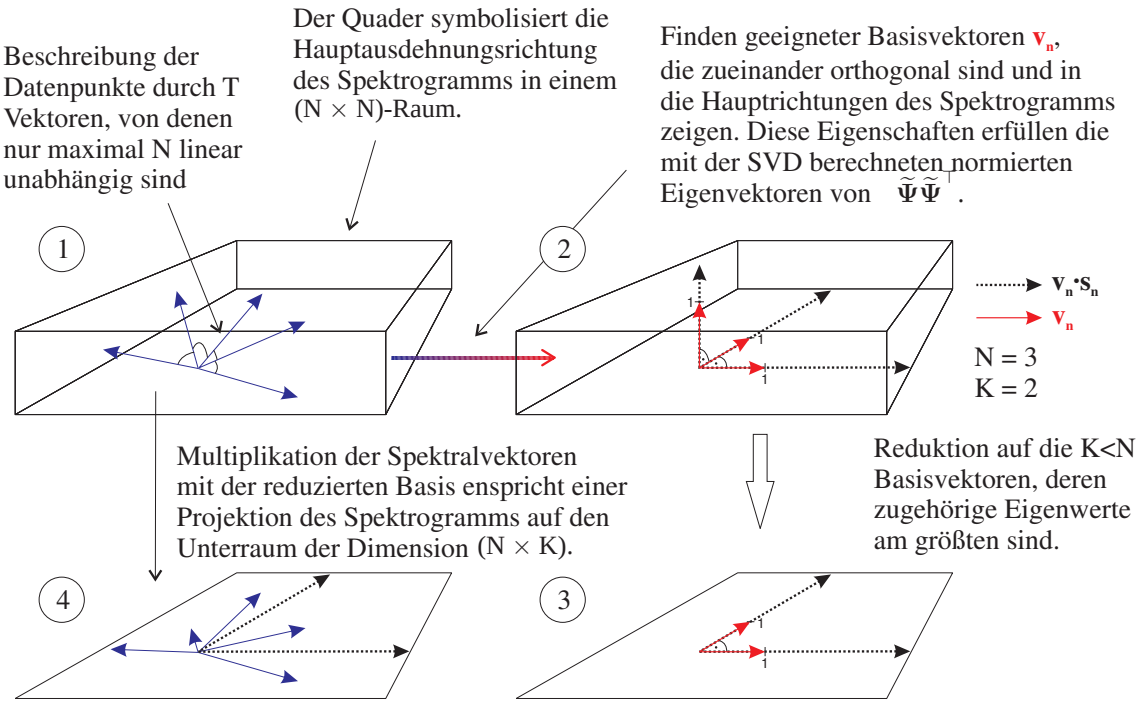


Abbildung 2.4: Prinzip der Dimensionsreduktion

2.2.2 Audio Spectrum Projection

Die eigentliche Reduktion des Spektrogramms findet in der Funktion „Audio Spectrum Projection“ statt. Dazu wird das Spektrogramm gemäß Gleichungen (2.9-2.12) von seiner Einhüllenden befreit. Anschließend wird dieses normalisierte Spektrogramm transponiert und mit der Transformationsmatrix $\mathbf{V}_{\mathrm{red}}$ multipliziert. Die Multiplikation entspricht einer *Projektion* der Daten auf die Basisvektoren des reduzierten Parameterraumes $\mathbf{V}_{\mathrm{red}}$.

$$\mathbf{P} \;=\; \widetilde{\mathbf{\Psi}}^{\top}_{(T\times N)} \cdot \mathbf{V}_{\mathrm{red}(N\times K)} \qquad \text{mit } \mathbf{P} \in \mathbb{R}^{T\times K} \tag{2.16}$$

Um später aus der Projektion $\mathbf{P}$ ein Spektrogramm rekonstruieren zu können, wird der Matrix $\mathbf{P}$ ein Spaltenvektor vorangestellt, der die Einhüllende (euklidische Norm) aus Gl. (2.11) beinhaltet. $\mathbf{P}$ wird im MPEG–7-Standard auf $\mathbf{P}_{\mathrm{erw}}$ erweitert:

$$\mathbf{P}_{\mathrm{erw}} \;=\; (\boldsymbol{\psi}_{\mathbf{env}}, \mathbf{P}) \qquad \Rightarrow \qquad \mathbf{P}_{\mathrm{erw}} \in \mathbb{R}^{T\times(K+1)} \tag{2.17}$$

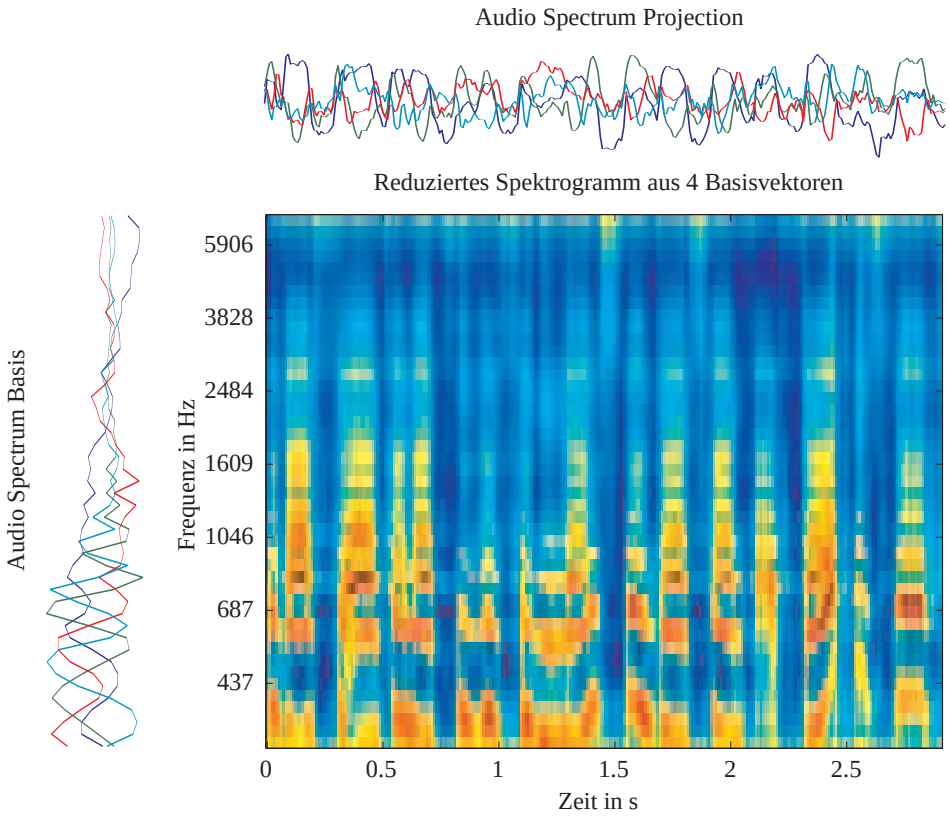


Abbildung 2.5: Rekonstruiertes Spektrogramm aus Audio Spectrum Basis und Audio Spectrum Projection

Die vollständige Rekonstruktion des Spektrogramms ist nicht möglich, wenn die Basisvektormatrix $\mathbf{V}$ reduziert wurde. Dennoch kann man ein Spektrogramm der Dimension $\mathbb{R}^{N\times T}$ berechnen, das nun aber von den „schwachen Eigenwerten“ befreit ist. Der Informationsverlust ist dabei sehr gering, wie man auch am Vergleich der Spektrogramme in Abb. 2.3 (original) und 2.5 (reduziert) erkennen kann. Hier ist das Spektrogramm auf nur 4 Basisvektoren reduziert, was einer Datenkompression von nahezu 10:1 entspricht. Die Berechnung des rücktransformierten reduzierten Spektrogramms erfolgt durch:

$$\mathbf{\Psi_{red}}(t) \quad = \quad \boldsymbol{\psi}_{\mathbf{env}}(t) \cdot \mathbf{P_t} \cdot \mathbf{V}_{\mathrm{red}}{}^{\top} \qquad t = 1 \ldots T \tag{2.18}$$

Für die Klassifikation ist die Rekonstruktion eines Spektrogramms nicht erforderlich. Auch die zu diesem Zweck gespeicherte Einhüllende $\boldsymbol{\psi}_{\mathbf{env}}$ enthält keine nützliche Information, da über die Signalleistung keine klassentypischen Eigenschaften beschrieben werden können. Sie fließt daher nicht in die Berechnung der Statistik ein, sondern dient lediglich der Normalisierung der Eingangsdaten $\mathbf{P}$ (auch Observationsdaten genannt).

In Abb. 2.4 kann man sehen, dass das reduzierte Spektrogramm als Projektion der Daten auf die Hauptrichtungsvektoren (oder Basis) aufgefasst werden kann, aus diesem Grunde wurde die Bezeichnung „Audio Spectrum Projection“ für diese Funktion der „MPEG-7 Multimedia Description Schemes“ [ISO02] gewählt.

Besonderheiten im Bezug auf die Klassifikation

Im Bezug zur Klassifikation ergibt sich bei den ASB/ASP im Vergleich zu anderen Merkmalen eine Besonderheit. Die Audio Spectrum Basis wird immer für die Gesamtheit der Spektrogramme aller Trainingsdaten einer Klasse berechnet. Dazu werden die ASE-Daten in zeitlicher Richtung aneinander gereiht, so als ob es sich um ein entsprechend langes Signal handelte. Dies ist notwendig, da mit der ASB bereits eine Auswahl der stärksten Komponenten der Spektren einer Klasse erfolgen soll. Die ASP wiederum kann mit Hilfe der klassenspezifischen Basisvektormatrix $\mathbf{V}_{\mathrm{red}}$ auch stückweise berechnet werden. Dabei ist zu beachten, dass ein Signal in jeder Klasse andere Projektionsdaten erzeugt. Die mit der ASB berechnete Basisvektormatrix $\mathbf{V}$ ist in diesem Zusammenhang als ein klassenspezifisches Koordinatensystem zu betrachten. Um Daten miteinander zu vergleichen, ist es immer notwendig, sie in das gleiche Koordinatensystem zu transformieren. Die Projektionsdaten P sind also die Transformation der ASE-Spektraldaten in das (auf die Hauptrichtungen reduzierte) Koordinatensystem der Klasse. Da $\mathbf{V}$ somit ein fester Bestandteil einer Signalklasse ist, muss diese Matrix auch für die Identifikation der Testsignale bekannt sein und wird daher zusammen mit den Modellparametern[3], die von den Klassifikationsverfahren berechnet werden, abgespeichert.
Aufgrund der Reduktion $\mathbf{V} \rightarrow \mathbf{V}_{\mathrm{red}}$ von N auf K Basisvektoren wird bei der Berechnung der Projektionsdaten bereits eine Art Filterung, bzw. eine Einschränkung der Freiheitsgrade, vorgenommen, deren (vorteilhafte) Wirkung auf die Klassifikation mit Simulationen untersucht wird. Vorab ist schon einmal in Abb. 2.6 aufgeführt, welche Auswirkungen durch die Dimensionsreduktion erwartet werden.

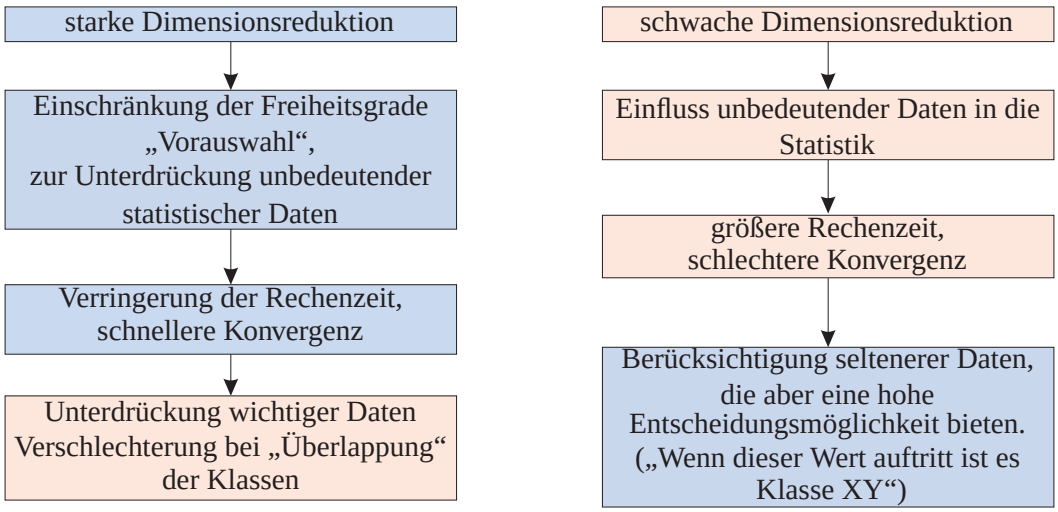


Abbildung 2.6: Vor- und Nachteile der Dimensionsreduktion

[3]Die Definition der Begriffe aus der Klassifikation folgt in Kapitel 3.

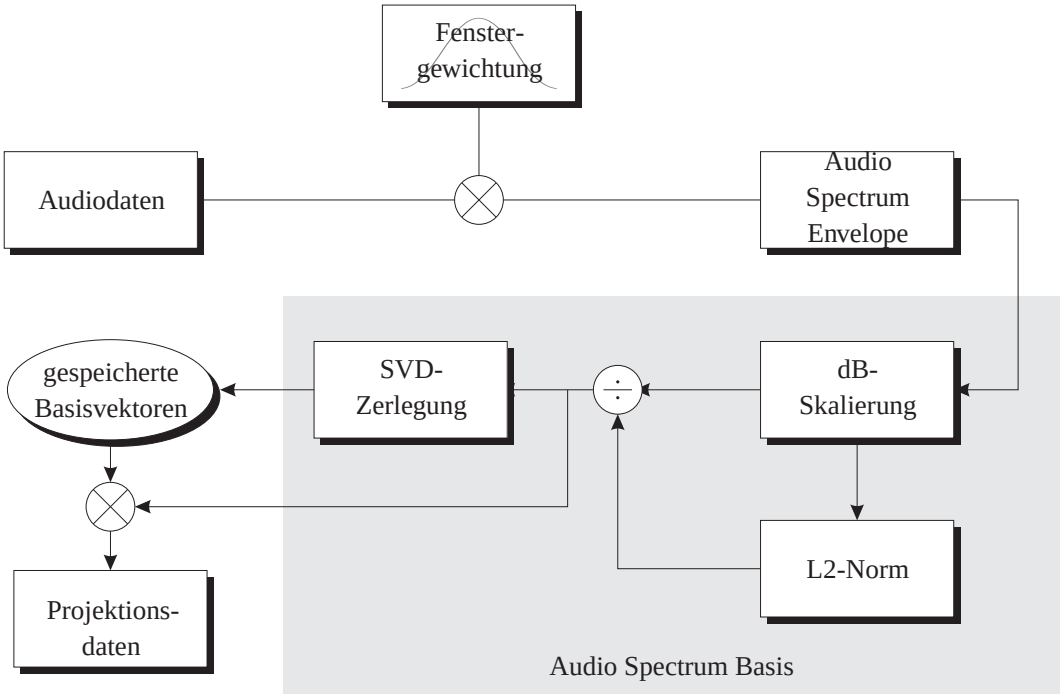


Abbildung 2.7: Extraktionsverfahren der MPEG-7 Merkmalvektoren

In Abb. 2.7 wird noch einmal der gesamte Ablauf der MPEG-7 Merkmalextraktion in einem Blockdiagramm veranschaulicht. Die Verarbeitung bezieht sich nur auf die Extraktion der *Trainingsdaten* einer Signalklasse. Für die Generierung der *Testdaten* entfällt die SVD-Zerlegung. Hier wird auf die gespeicherten Basisfunktionen der Klasse zurückgegriffen, gegen die das Signal getestet werden soll. Nähere Informationen hierzu folgen im Kapitel 3.

2.3 MFCC

Die im Bereich der Sprach- und der Sprechererkennung am weitesten verbreiteten Merkmale sind die Mel-Cepstrum-Koeffizienten (MFCC). Ein Cepstrum ist die inverse Fouriertransformierte (IDFT) eines logarithmierten Spektrums. Für das reelle, lineare Cepstrum ergibt sich:

$$Cep_s(k) \;=\; \mathrm{IDFT}\left\{\ln|S(n)|\right\} = \frac{1}{N}\sum_{n=0}^{N-1} S(n) W_N^{-kn} \tag{2.19}$$

$$\begin{array}{lccl} & S(n) & \bullet\!\!-\!\!\circ & s(k) \quad \text{(diskretes Zeitsignal)} \\ \text{mit} & S(n) & = & \mathrm{DFT}\left\{s(k)\right\} = \sum_{k=0}^{N-1} s(k) W_N^{kn} \\ & W_N^{kn} & = & e^{-j2\pi/N} \qquad \text{komplexer Drehoperator} \end{array} \tag{2.20}$$

Zur Berechnung der MFCC wird ein so genanntes „Mel-Spektrum“ verwendet, das mit Hilfe der nichtlinearen Mel-Filterbank aus dem linearen Spektrum erzeugt wird. Diese Mel-Filterbank besteht aus Dreieck-Bandpässen, die einander überlappen und näherungsweise logarithmisch über das Frequenzband verteilt sind. Die Mel-Skala kam ursprünglich durch ein Experiment

von Stevens und Volkmann (1940) zustande und basiert auf die subjektive Wahrnehmung von Frequenzvervielfachungen durch den Menschen. Da sie weitgehend einen logarithmischen Verlauf hat, wird die Mel-Skala durch folgende Funktion angenähert (siehe Abb.2.8):

$$f_{\text{Mel}} = 1127.01048 \cdot \ln\left(1 + \frac{f_{\text{Hz}}}{700}\right). \tag{2.21}$$

Die Dreieck-Bandpässe mit der Bandbreite b_m können, in Abhängigkeit zur Bandmittenfrequenz $U_{B_{MF}}$, durch folgenden Zusammenhang beschrieben werden:

$$U_{B_{MF}}(n) = \begin{cases} 1 - |n|/b(m) & |n| < b(m) \\ 0 & \text{sonst} \end{cases} \tag{2.22}$$

Anstelle der Berechnung der Bandmittenfrequenzen $B_{MF}(m)$ und zugehörigen Bandbreiten $b(m)$ aus Gl. (2.21) werden meistens die Werte aus Tabelle 2.1 verwendet:

Index m	1	2	…	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
$B_{MF}(m)$	100	200	…	900	1000	1149	1320	1516	1741	2000	2297	2293	3031	3482	4000	4595	5278	6063	6964
$b(m)$	100	100	…	100	124	160	184	211	242	278	320	367	422	484	556	636	734	843	969

Tabelle 2.1: Bandmittenfrequenzen und Bandbreiten der Mel Skala in Hz

Die wahrnehmungsgewichteten Werte an den Ausgängen dieser mel-skalierten Filter lassen sich durch Gl. (2.23) angeben:

$$Y_{\text{mel}}(m) = \sum_{k=B_{MF}(m)-b(m)}^{B_{MF}(m)+b(m)} S(n) U_{B_{MF}}(n + b(m)) \tag{2.23}$$

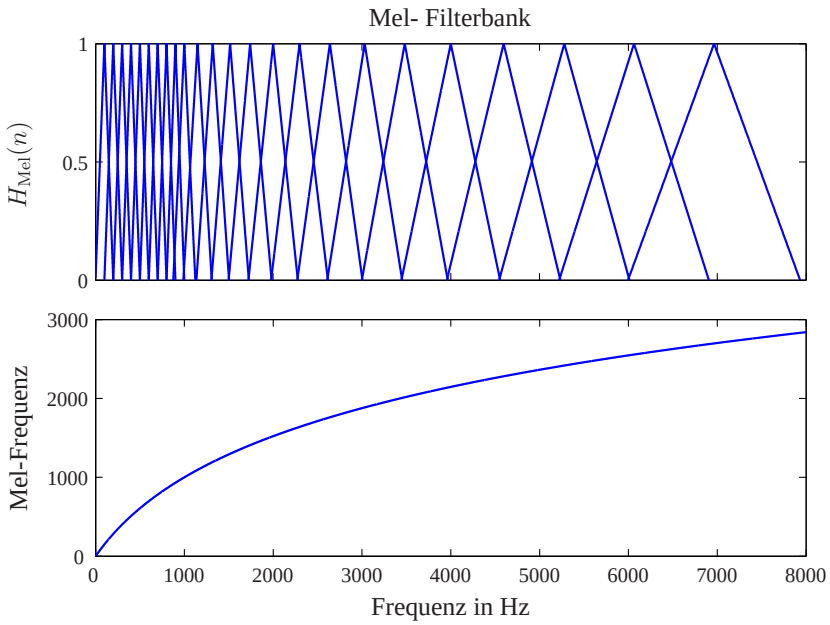


Abbildung 2.8: Mel-Filterbank

Die Berechnung des Mel-Cepstrums könnte nun nach Gl. (2.19) erfolgen. Da dieses Cepstrum reell ist, verwendet man in der Praxis auch häufig die inverse diskrete Cosinustransformation (IDCT). Die IDCT braucht weniger Rechenleistung als die IDFT, da die Multiplikationen der komplexen Signalanteile wegfallen. Ihre Koeffizienten sind ebenfalls stark unkorrelliert [VHH98]. Damit ergibt sich:

$$c(k) \quad = \quad \sum_{m=1}^{M} \ln |Y_{\mathrm{mel}}(m)| \cdot \cos \left(k \left(m - \frac{1}{2} \right) \frac{\pi}{M} \right) \tag{2.24}$$

Häufig wird zur Einebnung des Spektrums eine Filterung mit einem Hochpass-FIR-Filter erster Ordnung vorgenommen. Diese Vorverarbeitung nennt man auch Preemphase:

$$H(z) \quad = \quad 1 - a \cdot z^{-1} \tag{2.25}$$

Anschließend wird eine Blockzerlegung des Signals mit einer Hamming-Fenstergewichtung durchgeführt. Der gesamte Ablauf der Merkmalextraktion für die MFCC ist im Blockdiagramm 2.9 dargestellt.

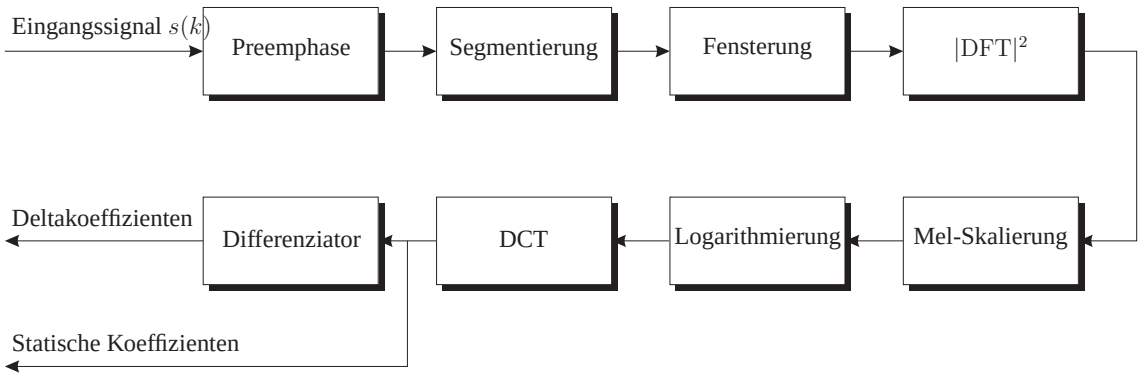


Abbildung 2.9: Block Diagramm der MFCC Merkmalgewinnung

Am Ende der Extraktion können zusätzlich zu den MFCC Deltakoeffizienten als Merkmal verwendet werden. Diese Differenzen aus den Mel-Cepstralkoeffizienten können besonders für höhere Frequenzen eine bessere Repräsentation der gewünschten Signaleigenschaften liefern. Die Deltakoeffizienten enthalten implizit eine Zeitinformation über den spektralen Verlauf (engl. spectral evolution). In den Simulationen wurden diese Delta-Merkmale nicht betrachtet.

Kapitel 3

Klassifikationsverfahren

Stochastische Signale, wie z.B. die Merkmalvektoren, lassen sich mit Hilfe von Wahrscheinlichkeitsverteilungen beschreiben. Die grundlegendste Verteilung ist dabei die Gaußsche Normalverteilung. Eine einzelne Normalverteilung, gegeben durch die Parameter Mittelwert μ und Varianz σ^2, reicht häufig nicht aus, um zu einer Signalklasse gehörende Merkmale zu beschreiben. Zu ähnlich ist der *globale* Mittelwert über eine Klasse von Merkmalvektoren zu einer anderen Klasse, als dass man dadurch eine eindeutige Zuordnung machen könnte. Bei allen Klassifikationsverfahren geht man daher von einer Mehrfachverteilung aus. Bei den Hidden Markov Models (HMM) wird angenommen, dass diese Verteilungen in einer zeitlichen Abfolge auftreten, während bei den Gaussian Mixture Models (GMM) die zeitliche Abfolge der Verteilungen beliebig ist. Auch beim Vektorquantisierer (VQ) bleibt die zeitliche Information unberücksichtigt, hier werden zwar keine Wahrscheinlichkeitsverteilungen approximiert, aber es werden ebenfalls mehrere Mittelwerte gesucht, die das stochastische Signal möglichst gut datenreduziert abbilden (siehe Abb. 3.1). Hierauf wird in den Beschreibungen der Klassifikationsverfahren noch genauer eingegangen.

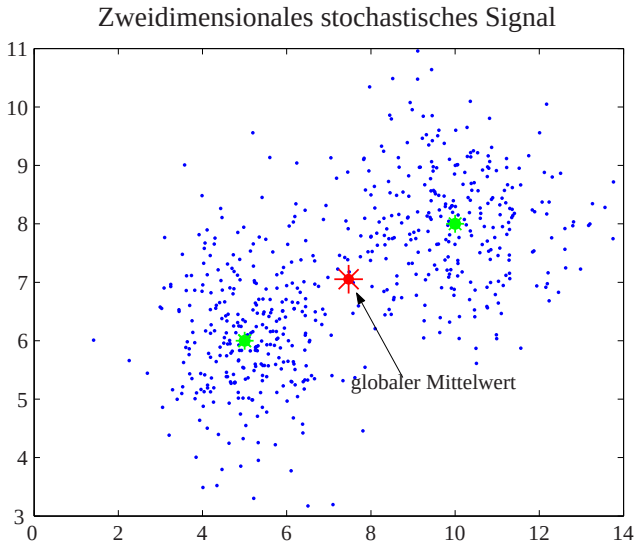


Abbildung 3.1: „Datenwolken“ (Cluster) eines zweidimensionalen stochastischen Signals

3.1 Gaussian Mixture Models

3.1.1 Überblick

Bei der Verwendung von Gaussian Mixture Models (GMM) zur Soundklassifikation geht man davon aus, dass die zeitliche Information in den Merkmalvektoren keine oder nur eine untergeordnete Rolle spielt. Daher reicht es aus, lediglich die Gesamtverteilungsdichte aller Merkmalvektoren einer Klasse zu betrachten. Die Merkmalvektoren können also beliebig zeitlich zu einem langen Observationsvektor aneinander gereiht werden. Dieser statistische Zufallsvektor wird zur Vereinfachung zunächst als Sequenz *skalarer* Merkmale definiert:

$$\mathbf{x} \quad = \quad [x_1, x_2, \ldots x_T]^\top . \tag{3.1}$$

Später erfolgt dann die Erweiterung auf mehrdimensionale Merkmale, wie z.B. Spektrogramm-Daten. Die Indizierung wurde aus Gründen der besseren Übersichtlichkeit gewählt und beschreibt den Wert des Observationsvektors im Analyseblock t:

$$x_t \quad = \quad x(t); \qquad t = 1 \ldots T \tag{3.2}$$

Wenn nicht anders definiert, sind alle Vektoren $\mathbf{x}$ Spaltenvektoren:

$$\mathbf{x} \quad \in \quad \mathbb{R}^{T \times 1}. \tag{3.3}$$

In Abb. 3.2 ist das Histogramm eines Merkmalvektors $\mathbf{x}$, z.B. eines MFCC Koeffizienten, zu sehen. Die Verteilungsdichte von $\mathbf{x}$ lässt sich meistens nur ungenau mit Hilfe einer einzelnen Normal- bzw. Gaußverteilung beschreiben.

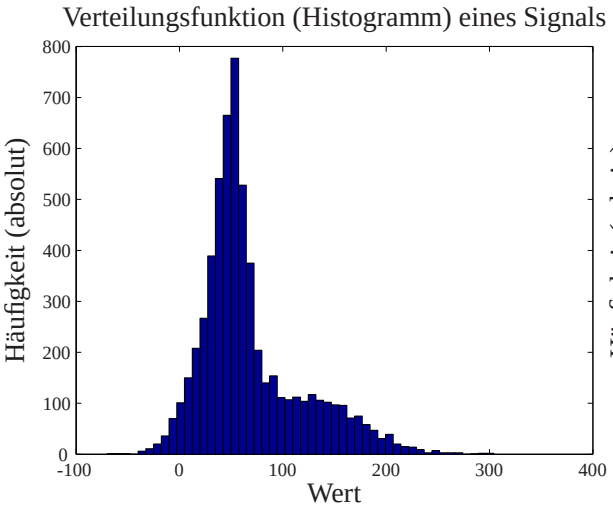


Abbildung 3.2: Histogram eines Signalvektors $\mathbf{x}$

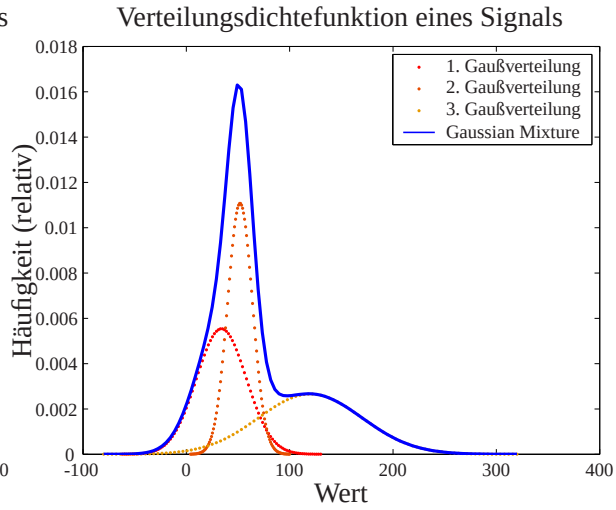


Abbildung 3.3: Approximation durch eine Gaußsche Mischverteilung (GM)

Eine genauere Approximation ist aber mit einer Gaußschen Mischverteilung möglich, die sich additiv aus mehreren einzelnen Gaußschen Normalverteilungen zusammensetzt [MP00] (siehe Abb. 3.3). Eine M-fache Gaußverteilung kann dabei vollständig mit $3M$ Parametern beschrieben werden:

$$\left.\begin{array}{lcll} p_i & : & \text{Mischungsgewichte} & \\ \mu_i & : & \text{Mittelwerte} & i = 1 \ldots M; \\ {\sigma_i}^2 & : & \text{Varianzen} & \end{array}\right\} = 3M \text{ Parameter} \tag{3.4}$$

Die Parameter werden zusammengefasst als

$$\lambda \quad = \quad \{p_i, \mu_i, \sigma_i^2\}; \quad i = 1 \ldots M \tag{3.5}$$

dargestellt.

Im englischen Sprachraum wird eine Gaußsche Mischverteilung als Gaussian Mixture Density oder kurz Gaussian Mixture (GM) bezeichnet. Eine GM ist die gewichtete Summe aus M Gaußverteilungen $b_i(X)$. Die Wahrscheinlichkeitsdichte einer GM wird durch folgende Gleichung beschrieben:

$$p(X|\lambda) \quad = \quad \sum_{i=1}^{M} p_i \cdot b_i(X) \tag{3.6}$$

$$b_i(X) \quad = \quad \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(X-\mu_i)^2}{2\sigma_i^2}} \tag{3.7}$$

mit

$$\sum_{i=1}^{M} p_i \quad = \quad 1. \tag{3.8}$$

Dabei ist X ein stochastischer Prozess, der unter Annahme der Ergodizität [Hän97] durch den Zufallsvektor $\mathbf{x}$ ersetzt werden kann. Diese Annahme ist (näherungsweise) dann erfüllt, wenn

von einer Signalklasse ausreichend statistische Daten zur Verfügung stehen.

Für die Signalklassifikation wird eine Klasse durch eine Gaußsche Mischverteilung bestimmt und durch den Modellparameter λ repräsentiert. In Abb. 3.4 ist die Zusammensetzung einer GM aus M gewichteten Gaußverteilungen $b(X)$ veranschaulicht.

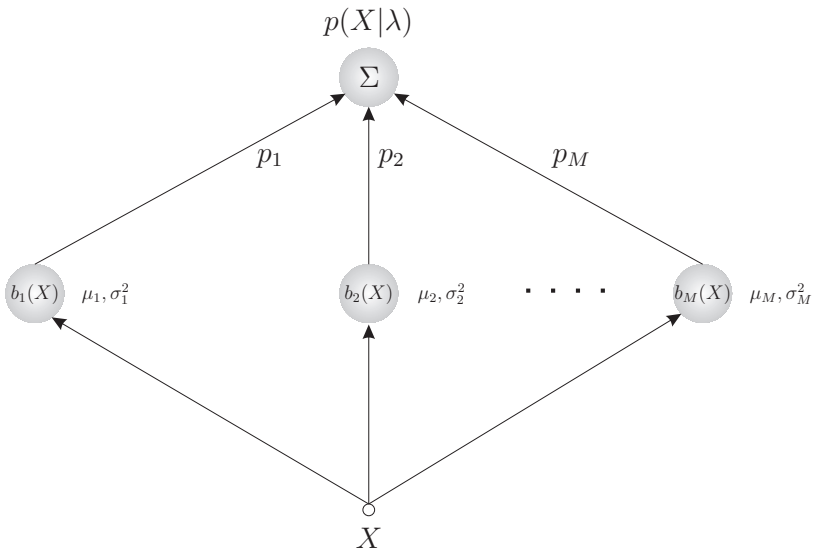


Abbildung 3.4: Zusammensetzung einer M-fachen Gaußschen Mischverteilung (GM).
Dabei bezeichnen $P(X|\lambda)$ die GM-Verteilung, $p_i, i = 1, \ldots, M$ die Mischungsgewichte
und $b_i(X), i = 1 \ldots, M$ die einzelnen Gaußverteilungen mit ihren Parametern μ_i und σ_i^2

Für die Approximation eines Gaußschen Mischverteilungsmodells (GMM) wird häufig ein Spezialfall des „Expectation-Maximization“ (EM)-Algorithmus von Dempster, Laird und Rubin [DLR77] verwendet, der im folgenden Abschnitt beschrieben wird. Ist das Klassenmodell λ bestimmt, so kann mit ihm die Klassifikation eines Signals erfolgen (siehe Abschnitt 3.1.4).

3.1.2 Der Expectation-Maximization (EM)-Algorithmus

Die grundsätzliche Idee des EM-Algorithmus ist es, iterativ durch eine abwechselnde Klassifikation (Expectation, E-Schritt) und eine anschließende Anpassung der Modellparameter $\lambda = \{p_i, \mu_i, \sigma_i^2\}$ (Maximization, M-Schritt), die Wahrscheinlichkeit für das Auftreten eines stochastischen Prozesses X bei gegebenen Modell λ zu maximieren. Unter Annahme der Ergodizität (d.h. $\mathbf{x}$ ist repräsentativ für X) gilt:

$$p(X|\lambda) \quad \underbrace{=}_{\text{Ergodizität}} \quad p(\mathbf{x}|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \stackrel{!}{=} \text{Max.} \tag{3.9}$$

Da der stochastische Prozess X durch die Trainingsdaten $\mathbf{x}$ der Klasse gegeben ist, müssen zur Maximierung die Modellparameter λ angepasst werden. Die Voraussetzung für das Auffinden dieses Maximums ist, dass nach jedem Induktionsschritt und der Berechnung eines neuen Modells $\overline{\lambda}$ gilt:

$$p(X|\overline{\lambda}) \geq p(X|\lambda) \qquad \text{(Monotoner Anstieg der Wahrscheinlichkeit)}. \tag{3.10}$$

Dieses Verfahren wird fortgeführt, bis ein Konvergenzschwellwert erreicht ist.
Im Bezug auf die Signalklassifikation entspricht der EM-Algorithmus der *Trainingsphase* des Modells. Sind die Eingangsdaten Merkmalvektoren einer Klasse, die eine ähnliche Verteilungsdichte haben, so erhält man durch die iterative Maximum-Likelihood(ML)-Schätzung des EM ein repräsentatives Modell λ für diese Klasse.

Zur Initialisierung des Lernalgorithmus wird die Anzahl der überlagerten Gaußverteilungen M und ein GMM mit beliebigen Parametern $\lambda = \{p_i, \mu_i, \sigma_i^2\}$ gewählt.

Da es sich um eine zusammengesetzte Verteilungsfunktion (*engl*. mixture) handelt, kann die Wahrscheinlichkeit für das Auftreten eines Datums x_t nicht direkt bestimmt, sondern muss für jede einzelne Gaußverteilung berechnet werden. Die Gesamtwahrscheinlichkeit ist dann die Summe aus allen gewichteten Einzelwahrscheinlichkeiten $b_i(\cdot)$:

$$p(x_t|\lambda) = \sum_{i=1}^{M} p_i \cdot b_i(x_t) \tag{3.11}$$

$$b_i(x_t) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x_t-\mu_i)^2}{2\sigma_i^2}} \tag{3.12}$$

Zur Optimierung des Modells wird eine Aussage darüber benötigt, wie gut eine *einzelne* Gaußverteilung i zur Gesamtverteilung von $\mathbf{x}$ passt. Dies wird erreicht, indem die Wahrscheinlichkeit von x_t im Bezug auf eine einzelne Gaußverteilung i durch die Wahrscheinlichkeit von x_t im Bezug auf die Gesamtverteilung (GM) dividiert wird, so dass man folgende a-posteriori-Wahrscheinlichkeit erhält (Expectation, E-Schritt) (siehe Abb. 3.5):

$$p(i|x_t, \lambda) = \frac{p_i b_i(x_t)}{\sum_{l=1}^{M} p_l b_l(x_t)} \tag{3.13}$$

Den Mittelwert aus einer großen Anzahl der Wahrscheinlichkeiten $p(i|x_t, \lambda)$ kann man zur Gewichtung der einzelnen Gaußverteilungen verwenden (Gl. 3.14). Entsprechend ergibt sich die Anpassung der anderen Parameter nach den Gleichungen (3.14-3.16) (M-Schritt):

Mischungsgewichte:

$$p_i = \frac{1}{T} \sum_{t=1}^{T} p(i|x_t, \lambda) \qquad i = 1, \ldots, M \tag{3.14}$$

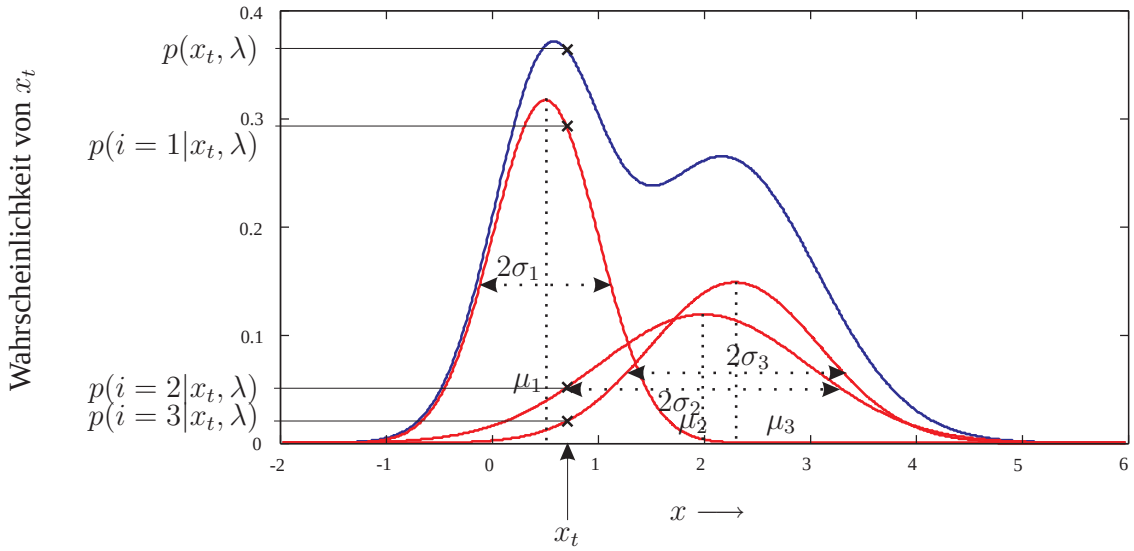


Abbildung 3.5: A-posteriori-Wahrscheinlichkeit

Mittelwerte:

$$\mu_i = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) \cdot x(t)}{\sum_{t=1}^{T} p(i|x_t, \lambda)} \qquad i = 1, \ldots, M \tag{3.15}$$

Varianzen:

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) \cdot x^2(t)}{\sum_{t=1}^{T} p(i|x_t, \lambda)} - \mu_i^2 \qquad i = 1, \ldots, M \tag{3.16}$$

Nach der Neuberechnung des Modells λ folgt erneut eine Klassifikation (bzw. Expectation, E-Schritt) mit anschließender Maximierung (M-Schritt). Dieser Vorgang wird iterativ fortgeführt, bis ein Konvergenzkriterium erfüllt ist und damit das GMM ausreichend die Verteilung des Eingangssignals approximiert oder bis die maximale Anzahl der zuvor festgelegten Iterationen erreicht ist.

Erweiterung auf mehrdimensionale Merkmalvektoren

Der vorangegangene Algorithmus wurde bisher nur für eindimensionale Merkmalvektoren $\mathbf{x}$ definiert. In der Regel braucht man für die Klassifikation von Signalen mehrere Merkmalvektoren oder mehrdimensionale Merkmale. Daher müssen die Gleichungen auf mehrdimensionale (multivariate) Gaußverteilungen erweitert werden. Der Vektor $\mathbf{x}$ ist nun eine Matrix $\mathbf{X}$ der Dimension $T \times N$. T ist weiterhin die Anzahl der zeitlich nacheinander erfolgten Einzelobservationen (T=Zeitdimension). N ist die Summe der Dimensionen in Parameterrichtung (z.B. Anzahl Cepstralkoeffizienten) und M Anzahl der Gaußverteilungen oder die Ordnung des Mischverteilungsmodells. Daraus resultiert auch die Änderung der Dimension der Mittelwerte und der Übergang von Varianzen hin zu Kovarianzmatrizen:

$$\begin{aligned}
\mathbf{x} \Rightarrow \mathbf{X} &\in \mathbb{R}^{(T\times N)} \\
x_t = x(t) \Rightarrow \mathbf{x_t} = \mathbf{x}(t) &\in \mathbb{R}^{(1\times N)} \\
\mu \Rightarrow \boldsymbol{\mu} &\in \mathbb{R}^{(M\times N)} \\
\mu_i \Rightarrow \boldsymbol{\mu_i} &\in \mathbb{R}^{(1\times N)} \\
\sigma \Rightarrow \mathbf{C_{xx}} &\in \mathbb{R}^{(M\times N\times N)} \\
\sigma_i \Rightarrow \mathbf{C_{xx_i}} &\in \mathbb{R}^{(1\times N\times N)} \\
p_i &\in \mathbb{R}^{(M\times 1)}
\end{aligned} \tag{3.17}$$

Die mit i indizierten Werte sind die Parameter der i-ten Komponente der Mischverteilung $i = 1 \ldots M$. Bei einer N-dimensionalen Gaußverteilung (engl. N-variate Gaussian Density) wird Gleichung (3.7) zu:

$$b_i(\mathbf{x_t}) = \frac{1}{2\pi^{N/2} \cdot |\mathbf{C_{xx_i}}|^{1/2}} \cdot \exp\left\{-\frac{1}{2} \cdot (\mathbf{x_t} - \mu_i) \cdot \mathbf{C_{xx_i}}^{-1} \cdot (\mathbf{x_t} - \mu_\mathbf{i})^\top\right\} \tag{3.18}$$

Dabei ist b_i ein skalarer Wert, der die Wahrscheinlichkeit für das Auftreten des Vektors $\mathbf{x_t}$ bei gegebenen Mittelwertvektor $\mu_\mathbf{i}$ und der Kovarianzmatrix $\mathbf{C_{xxi}}$ angibt.

Durch die Änderung der Dimension verändert sich auch die Berechnung der a-posteriori-Wahrscheinlichkeit und der Modellparameter:

A-posteriori-Wahrscheinlichkeit:

$$p(i|\mathbf{x_t}, \lambda) = \frac{p_i b_i(\mathbf{x_t})}{\sum_{l=1}^{M} p_l b_l(\mathbf{x_t})} \tag{3.19}$$

Mischungsgewichte:

$$p_i = \frac{1}{T}\sum_{t=1}^{T} p(i|\mathbf{x_t}, \lambda) \qquad i = 1, \ldots, M \tag{3.20}$$

Mittelwerte:

$$\mu_\mathbf{i} = \frac{\sum_{t=1}^{T} p(i|\mathbf{x_t}, \lambda) \cdot \mathbf{x_t}}{\sum_{t=1}^{T} p(i|\mathbf{x_t}, \lambda)} \qquad i = 1, \ldots, M \tag{3.21}$$

Kovarianzmatrizen:

$$\mathbf{C_{xx_i}} = \frac{\sum_{t=1}^{T} p(i|\mathbf{x_t}, \lambda) \cdot \mathbf{x_t}^T \mathbf{x_t}}{\sum_{t=1}^{T} p(i|\mathbf{x_t}, \lambda)} - \mu_\mathbf{i}^T \mu_\mathbf{i} \qquad i = 1, \ldots, M \tag{3.22}$$

3.1.3 Anschauliche Beschreibung der GM-Modellberechnung

Aus den Gleichungen (3.19) bis (3.21) ist nicht sofort einsichtig, warum es sich bei der iterativen Modellberechnung um ein Expectation-Maximization Verfahren handelt. Dies liegt daran, dass diese bereits eine Maximum-Likelihood Lösung des Optimierungsproblems $\frac{\partial p(X|\lambda)}{\partial \lambda} = 0$ darstellen.

Eine formale Herleitung des EM für Mischverteilungen findet sich in [MP00] und soll hier nicht durchgeführt werden. Stattdessen wird versucht zu veranschaulichen, warum die angegebenen Schritte zu einer iterativen Modelloptimierung führen. Dazu betrachten wir die Gaußsche Mischverteilung aus Abb. 3.5. Eine Komponentenverteilung i der GM kann man auch als eine *Klasse im Signal* auffassen. Es handelt sich dabei nicht um die Signalklasse, die durch ein vollständiges GMM repräsentiert wird, sondern um bestimmte Frequenzen oder *spektrale Muster*[1], die *innerhalb* eines Signals (mit einer gaußverteilten Wahrscheinlichkeit) auftreten können. Die Wahrscheinlichkeit eines Datenwertes x_t zu dieser inneren Klasse i zu gehören, wird durch die Wahrscheinlichkeit der Komponentenverteilung i ausgedrückt. Bei der Modelloptimierung interessiert uns aber gerade der umgekehrte Fall, d.h. wie gut passt die i-te Komponentenverteilung zum Datum x_t? Eine Aussage hierüber liefert der Quotient aus der Komponentenwahrscheinlichkeit $p(i|x_t, \lambda)$ und der Wahrscheinlichkeit von x_t im Bezug auf die gesamte Mischverteilung, also der Summe aller Komponentenwahrscheinlichkeiten gemäß Gl. (3.13) bzw. (3.19). Dieser Wert wurde bereits als a-posteriori-Wahrscheinlichkeit definiert und entspricht dem Expectation-Schritt des EM-Algorithmus. Eine Aufzeichnung des zeitlichen Verlaufs könnte für die drei Komponenten aus Abb. 3.4 etwa so aussehen, wie in Abb. 3.12. Aus der a-posteriori-Wahrscheinlichkeit können nun verschiedene Informationen für die Maximierung gewonnen werden.

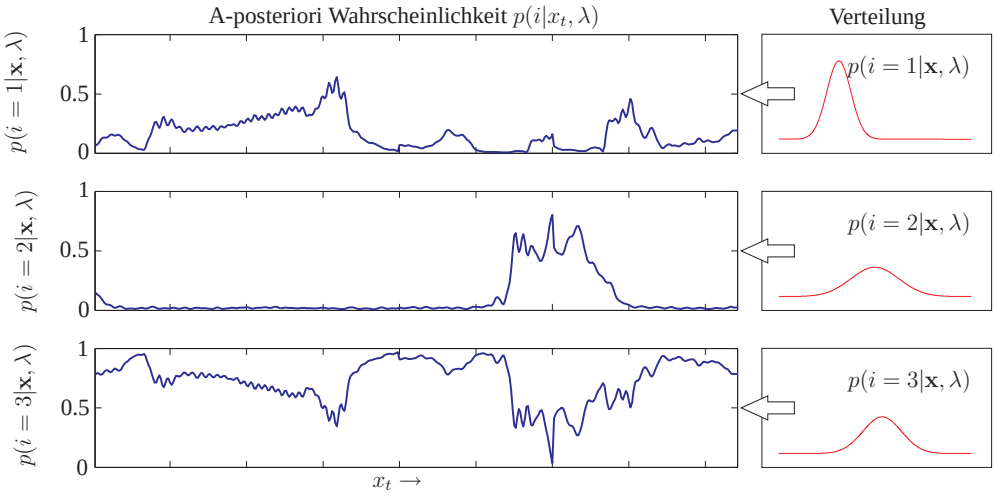


Abbildung 3.6: A-posteriori-Wahrscheinlichkeit

[1] immer vorausgesetzt, man verwendet die spektralen Merkmaldaten aus Kapitel 2

Der Mittelwert der a-posteriori-Wahrscheinlichkeit einer Komponentenverteilung aus T Datenwerten x_t gibt offenbar die Häufigkeit der Komponente im Signal $\mathbf{x}$ an und wird daher als Mischungsgewicht p_i für das optimierte Modell $\overline{\lambda}$ verwendet. Der Erwartungswert einer diskreten Zufallsvariable ist in (B.3) definiert. Die Neuberechnung des Mittelwertes der i-ten Komponente kann somit durch das gewogene Mittel von X erfolgen, unter Berücksichtigung einer Normierung auf die Summe aller Gewichte, siehe Gl. (3.15). Damit ist sichergestellt, das nur Datenwerte, die in den „Einflussbereich“ der i-ten Komponente fallen, diese auch verschieben können. Man könnte auch von einer „weichen“ Zuordnung der Datenwerte x_t zu einer Komponente sprechen, wohingegen der Vektorquantisierer (in der Regel) eine „harte“ Zuordnung zu einem Codevektor macht und so nur die zugeordneten Datenwerte den Vektor auf ihr geometrisches Mittel verschieben können (siehe Kap. 3.3). Das gleiche gilt auch für die Optimierung der Varianzen in Gl. (3.16). Mit diesen Berechnungen ist ein Schritt in die Richtung der Maximierung der Modellparameter erfolgt. Um eine weitere Optimierung des Klassenmodells λ zu erzielen, muss dieses Verfahren iterativ fortgesetzt werden, also wieder die Berechnung der a-posteriori-Wahrscheinlichkeit (E-Schritt) und Verschiebung der Modellparameter (M-Schritt).
Da hiermit das GMM bereits vollständig bestimmt ist, ist klar, dass die Information, die im *zeitlichen* Verlauf der a-posteriori-Wahrscheinlichkeit steckt, für die GMM verloren ist. Die Information über die Übergangswahrscheinlichkeit von einer Komponente i in eine andere Komponente j kann aber über Markov-Ketten erfolgen. Dieses Verfahren wird bei den Hidden Markov Models angewandt.

3.1.4 Klassifikation

Für die Klassifikation eines Testsignals sind zunächst die Modellparameter λ aller betrachteten Klassen zu bestimmen. Es wird hier von einer „closed-set“-Klassifikation ausgegangen, d.h. jedes Testsignal wird genau einer Klasse zugeordnet; die Feststellung, dass ein Testsignal zu *keiner* der betrachteten Klassen gehört, ist demnach nicht möglich. Die Klassifikation eines Signals kann durch Bestimmung einer minimalen Distanz oder der Berechnung einer maximalen Wahrscheinlichkeit (Maximum-Likelyhood) im Bezug auf alle Modelle erfolgen. Bei der GMM-Berechnung werden bereits Wahrscheinlichkeiten für das Auftreten einer Observation $\mathbf{X}$ bei gegebenen Modell λ durch die Gleichung

$$p(\mathbf{X}|\lambda) \quad = \quad \prod_{t=1}^{T} p(\mathbf{x_t}|\lambda) \tag{3.23}$$

bestimmt. Daher liegt es nahe, für die Klassifikation eine Likelyhood-Schätzung zu verwenden. Hat man nun insgesamt S dieser Klassenmodelle $\lambda_k, 1 \leq k \leq S$, wählt man für das Testsignal die Klasse aus, deren a-posteriori-Wahrscheinlichkeit am größten ist:

$$\hat{S} = \arg \max_{1 \leq k \leq S} p(\lambda_k|\mathbf{X}) = \arg \max_{1 \leq k \leq S} \frac{p(\mathbf{X}|\lambda_k)p(\lambda_k)}{p(\mathbf{X})} \tag{3.24}$$

Der zweite Teil von (3.24) ergibt sich aus dem Satz von Bayes und dem Satz der vollständigen Wahrscheinlichkeit (siehe B.1). Setzt man voraus, dass alle Klassen mit der gleichen Wahrscheinlichkeit auftreten, ist $p(\lambda_k) = 1/S$ und kann, da hier nur der relative Schätzwert benötigt wird, weggelassen werden. Dasselbe gilt für die Auftrittswahrscheinlichkeit $p(\mathbf{X})$, sie ist für alle Klassen gleich. Die Bestimmung des wahrscheinlichsten Klassenmodells vereinfacht sich so zu:

$$\hat{S} \quad = \quad \arg \max_{1 \leq k \leq S} p(\mathbf{X}|\lambda_k) \tag{3.25}$$

Unter Verwendung des Logarithmus der Wahrscheinlichkeit und Annahme der Unabhängigkeit zwischen den Observationen[2], kann die Klassifikation unter Berechnung der folgenden Gleichung durchgeführt werden:

$$\hat{S} \quad = \quad \arg \max_{1 \leq k \leq S} \sum_{t=1}^{T} \log p(\mathbf{x_t}|\lambda_k) \tag{3.26}$$

$$\text{mit} \qquad p(\mathbf{x_t}|\lambda) \quad = \quad \sum_{i=1}^{M} p_i b_i(\mathbf{x_t}) \tag{3.27}$$

3.1.5 Abgrenzung zu anderen Klassen

Die Zuverlässigkeit einer Klassifikation hängt davon ab, wie groß der Abstand zwischen den Klassenmodellen im Wahrscheinlichkeitsraum ist. Je höher dessen Dimension ist ($\widehat{=}$ Dimension der Observationen), umso weiter können die Mischverteilungen der Klassen (prinzipiell) voneinander entfernt liegen. Ein weiterer, wichtiger Faktor für die Diskrimination zwischen den Klassen ist die Varianz der einzelnen Gaußverteilungen eines Modells. Eine geringe Varianz ermöglicht eine hohe Abgrenzung zu anderen Klassen, während sich bei großen Varianzen eine starke Überlappung der Klassen ergibt. Die Größe der Varianz ist dabei auch davon abhängig, aus wie vielen Einzelverteilungen eine Mischverteilung zusammengesetzt ist.

[2] Diese Voraussetzung wurde bereits bei der Modellberechnung gemacht.

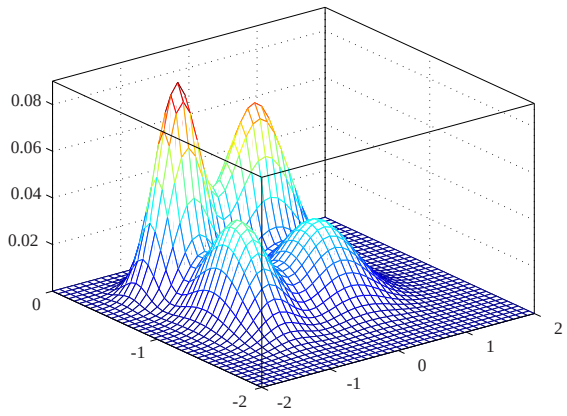


Abbildung 3.7: 2D-Mischverteilung

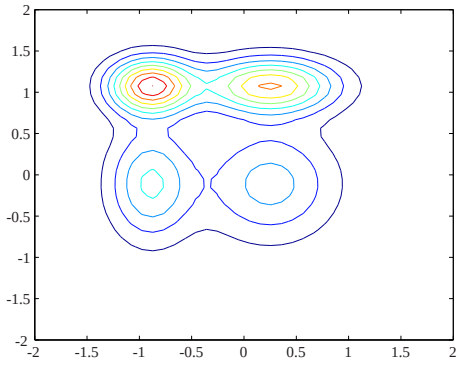


Abbildung 3.8: 2D-Mischverteilung, Konturdarstellung

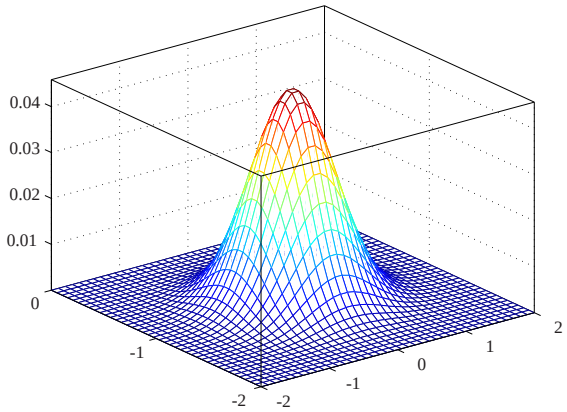


Abbildung 3.9: 2D-Gaußverteilung

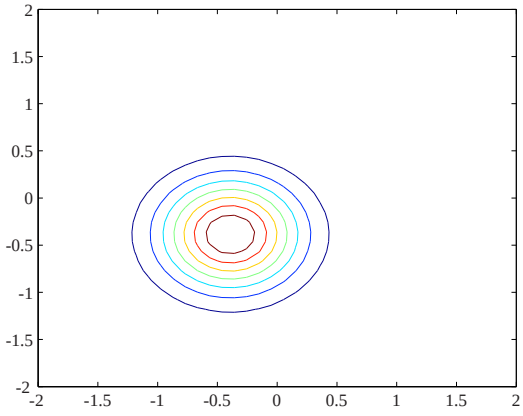


Abbildung 3.10: 2D-Gaußverteilung, Konturdarstellung

Dies ist anhand einer zweidimensionalen Gaußschen Mischverteilung der Ordnung[3] $M = 2$ in Abb. 3.7 und 3.8 veranschaulicht. Wenn man nur eine einzelne Gaußverteilung für die Approximation der Verteilung der Klassendaten verwendet, führt dies zu einer größeren Varianz und damit einer schlechteren Abgrenzung zu anderen Klassen (siehe Abb. 3.9 und 3.10).

Wählt man hingegen eine zu große Ordnung der Mischverteilung, kann dies zu Problemen bei der Berechnung des Modells führen, wenn zu wenig Trainingsdaten der Klasse zur Verfügung stehen. (Zum Beispiel könnte die Varianz bei wenigen Daten für eine Komponente unendlich klein werden, was zu Singularitäten in der inversen Kovarianzmatrix führt; eine Begrenzung der minimalen Varianz wiederum ist ein Eingriff in die Approximation des Modells.) Außerdem führt eine größere Modell-Ordnung zu einer Erhöhung des Rechenaufwands, sowohl bei der Modellberechnung als auch bei der Klassifikation. Eine Untersuchung der Modell-Ordnung im Bezug auf die Trefferquote wurde von D.A. Reynolds in [RR95] durchgeführt. Sie ergab, dass eine zu hohe Modell-Ordnung über $M > 16$ zu einer Verschlechterung der Trefferquote führte,

[3] mixture-order: Anzahl der Einzelverteilungen

wenn nicht ausreichend Trainingsdaten vorhanden waren.

3.1.6 Motivation für die Verwendung von vollen Kovarianzmatrizen

Ein GMM kann unterschiedlicher Gestalt sein, abhängig von der Art der Kovarianzmatrix. Das Modell kann eine Kovarianzmatrix pro Komponentenverteilung (lokale Kovarianz), eine Kovarianzmatrix für alle Komponentenverteilungen (große Kovarianz) oder eine Kovarianzmatrix für alle Sprechermodelle[4] haben (globale Kovarianz) [RR95]. Darüber hinaus kann eine Kovarianzmatrix voll oder diagonal sein. In dieser Arbeit werden nur lokale Kovarianzmatrizen betrachtet, also für jede Komponente i der Mischverteilung eine Kovarianzmatrix $\mathbf{C}_{\mathbf{xx_i}}$.
In den Simulationen in Kapitel 5 wird der Unterschied zwischen der Verwendung diagonaler und voller Kovarianzen untersucht. Die Motivation für die Verwendung voller Kovarianzmatrizen liegt darin, dass die einzelnen Dimensionen[5] der Observationsdaten in der Regel eine statistische Abhängigkeit zueinander haben. Diese Abhängigkeit lässt sich nicht durch diagonale Varianzen im Modell berücksichtigen. Die Einbeziehung der Korrelation der Observationsvektoren durch die Kovarianzmatrix kann also zu einer Verbesserung des Modells führen. Ob allerdings die dadurch gesteigerte Komplexität der Berechnung des Modells gerechtfertigt ist, muss für die jeweilige Anwendung der Klassifikation sowie auf Grundlage der verwendeten Observationen entschieden werden.
Die statistische Abhängigkeit der einzelnen Dimensionen der Observationsdaten ist anschaulich verständlich, wenn die Merkmale aus den Spektren harmonischer Signale extrahiert wurden. Leicht einzusehen ist z.B. bei dem Spektrogramm eines Sprachsignals, dass die Spektralverläufe der Harmonischen eine Abhängigkeit zu den Spektralverläufen der Grundfrequenz des Signals haben.
Auch bei der Verwendung eines orthogonalisierten (und reduzierten) Spektrogramms, den Audio-Spektrum-Projektionsdaten (ASP), ist lediglich eine *lineare* Unabhängigkeit zwischen den Observationsvektoren sichergestellt, es kann aber immer noch eine *statistische* Abhängigkeit zwischen den Daten geben.
Es gibt nun verschiedene Möglichkeiten, die statistische Abhängigkeit der Signale zu berücksichtigen. Man kann durch unabhängige Komponentenanalyse (engl. Independent Component Analysis, ICA) die statistisch unabhängigen Komponenten des Signals bestimmen. Dieser Ansatz wurde im Bezug auf die Klassifikation von Signalen auch von M. Casey [Cas01] untersucht. Algorithmen, wie man die unabhängigen Komponenten aus Signalen extrahiert, wurden unter anderem von M. Feng (JADE/MADE) [Fen99], von J.F. Cardoso (FastICA) [Car98] sowie von A. Hyvärinen (et al.) in [HKO01] vorgestellt. Ob sich Vorteile für die Klassifikation mit Hilfe unabhängiger Eingangsdaten ergeben, konnte im Rahmen dieser Arbeit nicht eingehender untersucht werden.

Eine andere, wesentlich einfachere Möglichkeit, ist die Berücksichtigung von Kovarianzen der Signale. In einer diagonalen Kovarianzmatrix stehen nur die Varianzen des Signals auf der

[4] in dem angegeben Literaturverweis wird nur die Sprecherklassifikation betrachtet.
[5] hiermit sind die Zeilenvektoren der $(N \times T)$-dimensionalen Observationsmatrix X gemeint

Diagonalen, alle anderen Werte (die Kovarianzen) sind Null. Verwendet man stattdessen die volle Kovarianzmatrix, erhält man zusätzlich zur Varianz noch Information über die Korrelation (und damit die statistische Abhängigkeit) der Signale in den einzelnen Dimensionen.

3.2 Hidden Markov Modelle

Das Hidden Markov Modell (HMM) ist ein zweistufiger stochastischer Prozess [Rab89]. Es besteht aus einer Markov-Kette[6] mit einer meist geringen Zahl von Zuständen, denen Wahrscheinlichkeiten bzw. Wahrscheinlichkeitsdichten zugeordnet sind. Mit diesen Zustandsketten ist es möglich, den zeitlichen Verlauf der Verteilung eines Signals zu beschreiben (siehe Abb. 3.11).

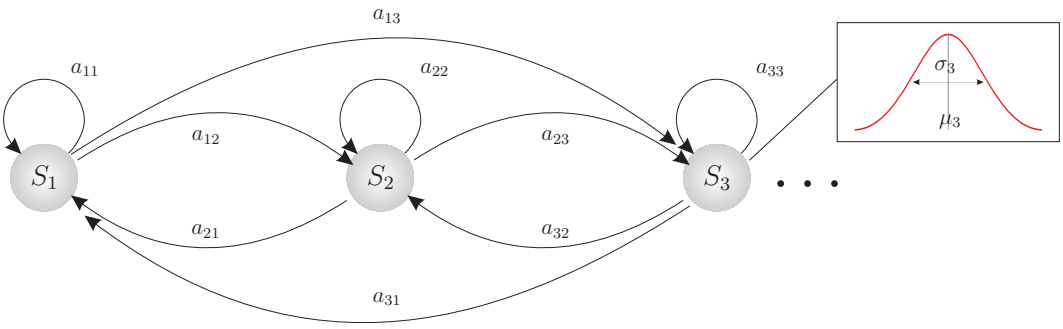


Abbildung 3.11: Markov-Kette mit 3 Zuständen

Der Übergang von einem Zustand S_i in einen Zustand S_j wird durch die Übergangswahrscheinlichkeit

$$a_{ij} \quad = \quad P(q_{t+1} = S_j | q_t = S_i), \qquad 1 \leq i, j \leq N. \tag{3.28}$$

beschrieben. Alle Übergangswahrscheinlichkeiten werden zu einer Matrix $\mathbf{A}$ zusammengefasst:

$$\mathbf{A} \quad = \quad \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix} \tag{3.29}$$

Außerdem wird für jeden Zustand einen Anfangswahrscheinlichkeit

$$\pi = (\pi_1, \pi_2, \dots, \pi_N) \tag{3.30}$$

definiert. Bei den hier verwendeten kontinuierlichen Hidden Markov Modellen (continuous markov models), werden die Zustände durch Wahrscheinlichkeitsverteilungen b_i approximiert.

[6] Die Markov Eigenschaft dieser Zustandsketten besagt, dass der Folgezustand nur vom gegenwärtigen Zustand abhängt und nicht von vorangegangenen Zuständen. Ein solches System bezeichnet man auch als „gedächtnislos“.

Im Bezug auf die Klassifikation werden hierfür meistens multivariate[7] Normalverteilungen gewählt:

$$b_i(\mathbf{x_t}) \quad = \quad \frac{1}{2\pi^{N/2} \cdot |\mathbf{C_{xx_i}}|^{1/2}} \cdot \exp\left\{-\frac{1}{2} \cdot (\mathbf{x_t} - \mu_\mathbf{i}) \cdot \mathbf{C_{xx_i}}^{-1} \cdot (\mathbf{x_t} - \mu_\mathbf{i})^\top\right\} \tag{3.31}$$

Zu den Dimensionierungen und Bezeichnungen siehe Gl. (3.17).
Die Zustände selbst sind verborgen („hidden“), da sie nicht direkt aus den Daten hervorgehen. Daher müssen die beobachtbaren Daten (observable data, Observationsdaten) verwendet werden, um aus ihnen auf die verborgenen Zustände zu schließen. Ein Zustand entspricht einem Daten-Cluster im Merkmalraum. In der Regel werden diese Datenwolken jedoch, aufgrund ihrer Komplexität, durch mehrere Zustände beschrieben. Die Anzahl dieser Zustände muss meistens empirisch und abhängig von den Merkmaldaten ermittelt werden, genau wie die Anzahl der Gauß-Komponenten beim GMM.

Klassenmodelle, die mit Hilfe der HMM erstellt werden, unterscheiden sich in den drei Parametern

- Zustandsübergangswahrscheinlichkeit $\mathbf{A}$,
- Wahrscheinlichkeitsverteilung eines Zustands $\mathbf{B} = (b_1, b_2, \ldots, b_N)$ (im Falle einer Normalverteilung gegeben durch Mittelwert und Varianz)
- und der Anfangswahrscheinlichkeiten π.

Man fasst sie in der Modellvariablen

$$\lambda \quad = \quad \{\mathbf{A}, \mathbf{B}, \pi\} \tag{3.32}$$

zusammen.
Ein stochastisches Signal $\mathbf{X}$ bezeichnet man im Bezug auf die Modellbildung auch als Observationssequenz. Nach [Rab89] ergeben sich für die HMMs drei große Problemstellungen:

1. Die Berechnung der Wahrscheinlichkeit für das Auftreten einer bestimmten Observationssequenz bei einem gegebenen Modell $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, $P(\mathbf{X}|\lambda)$.
2. Die Berechnung einer im Bezug auf die Observation $\mathbf{X}$ optimalen Zustandsfolge

   $$Q \quad = \quad (q_1, q_2, \ldots, q_T) \tag{3.33}$$

   bei einem gegebenen Modell λ, wobei q_t ein bestimmter Zustand $S_i \quad i = 1 \ldots M$ zum Zeitpunkt $t = 1 \ldots T$ ist.
3. Die Anpassung der Modellparameter λ um $P(\mathbf{X}|\lambda)$ zu maximieren.

[7] Das sind Normalverteilungen deren Zufallsvariable mehrdimensional ist, siehe Abschnitt 3.3.

Punkt 1) und 3) sind besonders für die Signalklassifikation von Bedeutung.
Prinzipiell ist es möglich, die genaue Wahrscheinlichkeit $P(\mathbf{X}|\lambda)$ für eine Observation $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$ zu bestimmen:

$$P(\mathbf{X}|\lambda) = \sum_{\text{alle}Q} P(\mathbf{X}|, Q, \lambda)P(Q, \lambda) \tag{3.34}$$

$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(\mathbf{x}_1) a_{q_1 q_2} b_{q_2}(\mathbf{x}_2) \ldots a_{q_{t-1} q_t} b_{q_t}(\mathbf{x}_T). \tag{3.35}$$

Allerdings müsste man dazu alle möglichen Zustandssequenzen Q betrachten, die von der Observation erzeugt werden können. Dies ist selbst für eine geringe Anzahl an Zuständen und kurzen Sequenzen mit einem enormen, untragbaren Rechenaufwand verbunden. Nach [Rab89] liegt die Anzahl der hierfür erforderlichen Multiplikationen bei $(2T - 1)M^T$ und die Anzahl der Additionen bei $M^T - 1$.
Bei $M = 5$ Zuständen und $T = 100$ Observationen währen also bereits $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ Rechenschritte erforderlich. Daher wird für diese Berechnung ein effizienter Algorithmus, der Forward-Backward-Algorithmus von L.E. Baum und J.A. Egon verwendet [BE67, BS68].

Auch für die Optimierung der Modellparameter aus Punkt 3) wird der Forward-Backward Algorithmus benötigt. Im Anschluss an die damit erzielte **Expectation** wird eine Maximierung der Modellparameter λ durchgeführt. Dieses EM-Verfahren wird iterativ fortgesetzt, bis ein Konvergenzkriterium erreicht ist. Für detaillierte Informationen zu der auch als *Baum-Welch-Verfahren* bekannten Schätzmethode, muss hier auf weiterführende Literatur verwiesen werden, z.B. [Rab89, BE67].

3.2.1 Klassifikation

Die Klassifikation einer Testsequenz erfolgt beim HMM über eine Viterbi-Dekodierung [Vit67]. Als Ausgang ist hier nur der Maximum-Likelihood-Wert (ML-Wert) der Observation $\mathbf{X}$ im Bezug auf die betrachtete Klasse von Bedeutung; die gleichzeitig durch den Viterbi-Algorithmus bestimmte optimale Zustandsfolge spielt für die Klassifikation keine Rolle. Die Testsequenz wird anschließend der Klasse zugeordnet, für die der Viterbi den größten ML-Wert ausgibt.

3.2.2 Der Unterschied zwischen GMM und HMM

Spektrale Struktur von Signalen

Bei Spracherkennern haben die Trainingsdaten einer Klasse in der Regel einen ähnlichen spektralen Verlauf. So kann z.B. ein bestimmtes Wort, das zum Training von vielen unterschiedlichen Sprechern gesprochen wird, anhand dieser spektralen Struktur erkannt werden. Ebenso verhält es sich mit Klängen, deren Charakteristik sich durch die zeitliche Entwicklung des Spektrums ausdrückt. Hundegebell hat z.B. oft einen typischen spektralen Verlauf in Form

einer Impulsfolge tieffrequenter Laute, während Applaus eher einen gleichmäßigen rauschartigen Charakter hat. Andere Klänge (z.B. von Musikinstrumenten) haben ebenfalls ein immer wiederkehrendes Ein- und Ausschwingverhalten, das sich zur Klassifikation ausnutzen lässt. Wenn diese spektralen Strukturen (z.B. Harmoniemuster) durch Zustandsfolgen ihrer Wahrscheinlichkeitsverteilungen modelliert werden (wie bei der HMM), kann man erwarten, dass u.a. durch die zeitliche Abfolge der Zustände eine Zuordnung zu einer Signalklasse möglich ist. Im obigen Beispiel ergibt sich beim Hundegebell ein ständiger Wechsel zwischen zwei oder drei Zuständen (Knurren - Bellen - Atmen - Knurren - Bellen), während gleichmäßiger Applaus längere Zeit in einem Zustand verharrt.
Es gibt aber auch viele Signale, bei denen der Zustandsverlauf keinen Rückschluss auf die Signalklasse zulässt. Bei einer großen Datenbank von männlichen und weiblichen Stimmen aus unterschiedlichen Aufnahmesituationen, mit unterschiedlichen Aufnahmelängen und Inhalten kann zum Beispiel die Abfolge bestimmter Frequenzen (bzw. harmonischer Spektralmuster) nicht zur Klassifikation ausgenutzt werden, wohl aber ihre statistische Häufigkeit. Das gleiche gilt für die Klassifikation von Musik verschiedener Stilrichtungen (z.B. zur Genreerkennung oder Gesangsdetektion) sowie für die Sprecherklassifikation. Derartige Klangklassen lassen sich nur durch die Wahrscheinlichkeitsverteilung ihrer Merkmale beschreiben, nicht aber über die Abfolge der Verteilung in Form einer Zustandsfolge.
Im ersten Fall kann das HMM, das eine Abfolge der Wahrscheinlichkeitsverteilungen mit Hilfe der Übergangsmatrix $\mathbf{A}$ beschreibt, von der typischen Struktur der Signale profitieren und liefert erwartungsgemäß mehr Information über die Signalklasse, als das GMM. Dies führt letztendlich zu einer besseren Diskriminationsfähigkeit und einer höheren Detektionsrate gegenüber dem GMM. Für die zweite Art von Signalen bringt das HMM keine Vorteile; hier liegt in den Zustandsübergängen a_{ij} aus Gl. (3.28) keine zusätzliche Information über die Klasse.
Die Modelloptimierung des HMM mit Hilfe des Baum-Welch-Algorithmus ist ähnlich zu der in Abschnitt 3.1.2 beschriebenen GMM-Optimierung. Der Ausgangspunkt ist auch hier die a-posteriori-Wahrscheinlichkeit einer Observationssequenz für einen Zustand i, $p(i|\mathbf{x}_t, \lambda)$ (siehe Abb. 3.12). In der aus diesen Daten berechneten Übergangsmatrix $\mathbf{A}$ sind implizit auch Gewichtungen der Zustände enthalten:

$$p_i = \frac{1}{M} \sum_{j=1}^{M} a_{ij} \qquad i = 1 \ldots M. \tag{3.36}$$

Diese normierten Spaltensummen von $\mathbf{A}$ können unter gewissen Bedingungen als Mischungsgewichte eines GMM interpretiert werden. Diese Bedingungen liegen dann vor, wenn die Observationsdaten keine zeitliche Struktur aufweisen, die Zustandsübergänge also zufällig und nur durch die Häufigkeit eines Zustands gewichtet sind. Für derartige Daten kann ein HMM immer noch die Informationen eines GMM liefern, dessen Komponentenverteilungen b_i den Zustandsverteilungen des HMM entsprechen. Der Nachweis für die Äquivalenz von HMM und GMM für zeitlich unstrukturierte Daten kann hier nicht geführt werden, da dies unter anderem eine genaue Analyse des Forward-Backward Algorithmus erfordern würde.

Mit dieser Annahme lässt sich aber erklären, warum das HMM-Verfahren für die betrachteten

Signalklassen keine besseren Modelle liefern kann, als ein GMM mit der gleichen Modell-Ordnung. Der größere rechnerische Aufwand des Baum-Welch-Algorithmus bei der Modellberechnung sowie des Viterbi-Algorithmus bei der Klassifikation/Identifikation ist für bestimmte Signalklassen also nicht gerechtfertigt.

In dieser Arbeit werden relativ umfangreiche Klangklassen ohne typische spektrale Zeitstruktur klassifiziert. Daher können mit dem GMM ebenso gute Detektionsraten erzielt werden, wie mit dem HMM. Da bei dem GMM eine geringere Anzahl an Parametern geschätzt werden muss, ist der EM-Lernalgorithmus des GMM etwas einfacher aufgebaut. Für diese Art von Signalklassifikationen scheint das Gaußsche Mischverteilungsmodell vorteilhafter zu sein.

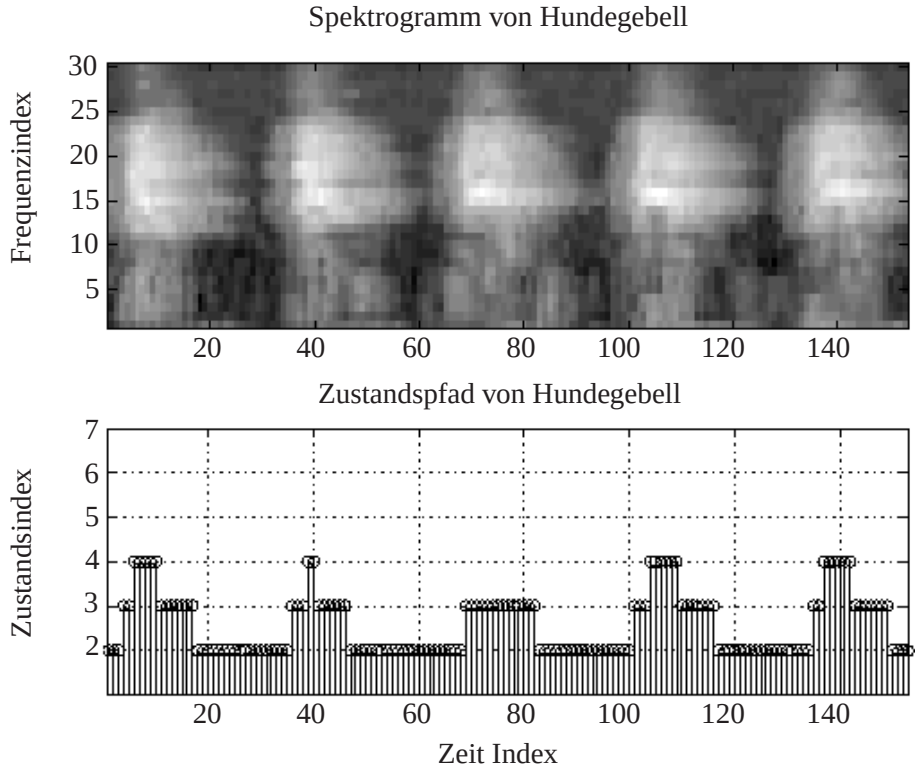


Abbildung 3.12: Audio Spectrum Envelope und Zustandspfad von Hundegebell mittels HMM

3.3 Vektorquantisierer

Vektorquantisierer sind die grundlegendste und anschaulichste Form aller Klassifikationsverfahren. Daher ist ihre Verbreitung in der Literatur auch entsprechend groß und es existieren sehr viele verschiedene Abwandlungen und Optimierungen für die jeweiligen Anwendungen. In dieser Arbeit soll nur exemplarisch die grundsätzliche Vorgehensweise beschrieben und ein Zusammenhang zu HMM und GMM hergestellt werden.

Quantisierung und Verzerrung Das Ziel einer Quantisierung ist immer eine Datenreduzierung eines Signals auf eine diskrete, begrenzte Anzahl an Quantisierungsstufen. Die Anzahl der zulässigen Datenwerte („Stufen“) ist in jedem Fall geringer, als die des Originalsignals. Eine Herabsetzung der Stufenzahl führt zwangsläufig zu einer Verzerrung des Signals. Als Maß für die Verzerrung eines quantisierten Signals kann man die mittlere Distanz eines Datenwertes zu der ihm zugeordneten Quantisierungsstufe verwenden. Eine Quantisierung muss nicht linear sein, das heisst die Quantisierungsstufen können so im Datenraum verteilt sein, dass sich im Bezug auf ein betrachtetes Signal bei einer festgelegten Anzahl an Quantisierungsstufen eine minimale Verzerrung ergibt. Anders ausgedrückt ist es das Ziel, eine möglichst gute Repräsentation für ein Signal mit begrenzter Anzahl an zulässigen Datenwerten zu erlangen.

Codevektoren und Codebuch Da besonders im Bezug auf die Klassifikation immer mehrdimensionale Signale betrachtet werden, wird ein zulässiger diskreter Quantisierungswert durch einen so genannten *Codevektor* beschrieben. Eine Ansammlung dieser Codevektoren wird zu einem *Codebuch* zusammengefasst. In Abb. 3.1 ist ein stochastisches Signal abgebildet, dass durch zwei Codevektoren (die großen Punkte im Schwerpunkt der Datenwolken) repräsentiert wird. Wenn nun eine Klasse von Daten (z.B. die Merkmalvektoren eines Sprechers) zu einer bestimmten kumulativen Häufung führt, kann man diese Klasse durch die Mittelpunkte dieser Datenwolken beschreiben. Diese Mittelpunkte (bzw. Schwerpunkte) sind die für die Signalklasse optimalen Quantisierungsstufen (Codevektoren) und können in Form eines Codebuchs abgespeichert werden.

Die Aufgabe des Vektorquantisierers ist dabei, diese optimalen Codevektoren zu finden. Wie bei den anderen Klassifikationsverfahren wird dies durch ein iteratives Optimierungsverfahren erreicht, bei dem abwechselnd die Verzerrung (mittlere Distanz zum Codevektor) berechnet (entspricht der *Expectation* beim EM) und anschließend die Verschiebung der Codevektoren (entspricht der *Maximization*) vorgenommen wird.

Konvergenz Als Distanzmaß wird meistens die quadratische euklidische Distanz verwendet. Daher kann auch beim Vektorquantisierer immer nur ein lokaler Extremwert (in diesem Fall die minimale Distanz eines Datenwerts zu einem Codevektor) gefunden werden. Das bedeutet, dass das Verfahren nicht zwangsläufig zum selben (absoluten, globalen) Optimum konvergiert und somit immer eine Restunsicherheit verbleibt. In solchen Fällen kann es vorkommen, dass sich

eine Signalklasse mit dem Verfahren nicht optimal durch die berechneten Codevektoren repräsentieren lässt und dies so zu Fehlklassifikationen der Testsignale führt. Bei einer geeigneten Initialisierung der Codevektoren, kann diese Unsicherheit aber gering gehalten werden.

Die Bestimmung der Verzerrung Die Verzerrung ist die mittlere Distanz aller Datenwerte zu den ihnen zugeordneten Codevektoren eines Codebuchs. Der Datenwert $\mathbf{x_t}$ sei ein N-dimensionaler Merkmalvektor zum Zeitpunkt $t, \quad \text{mit } t = 1 \ldots T$.
Das Codebuch $\mathbf{C} = (\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_M})$ enthält M Codevektoren der gleichen Dimension wie $\mathbf{x_t}$. Dann ergibt sich die quadratische euklidische Distanz des Vektors $\mathbf{x_t}$ im Bezug auf Codevektor $\mathbf{C_i}$ zu:

$$d(\mathbf{x_t}, \mathbf{C_i}) = \frac{1}{N} \sum_{n=1}^{N} |x_t(n) - C_i(n)|^2 \tag{3.37}$$

$$\text{mit } \mathbf{C_i} = [C_i(1), C_i(2), \ldots, C_i(N)]; \qquad \mathbf{x_t} = [x_t(1), x_t(2), \ldots, x_t(N)] \tag{3.38}$$

oder in vektorieller Schreibweise:

$$d(\mathbf{x_t}, \mathbf{C_i}) = \frac{1}{N} (\mathbf{x_t} - \mathbf{C_i})^\top (\mathbf{x_t} - \mathbf{C_i}) \tag{3.39}$$

Um zu berücksichtigen, dass die Distanzen in den einzelnen Dimensionen $1 \leq n \leq N$ sehr unterschiedlich sein können, wird meistens auf die Varianzen von $\mathbf{x_t}$ normiert. Ohne diese Normierung würden große Datenwerte mit großen Varianzen das Distanzmaß dominieren, so dass die Werte der anderen Dimensionen keinen großen Einfluss mehr hätten.

Normierte Euklidische Distanz:

$$d(\mathbf{x_t}, \mathbf{C_i}) = \frac{1}{N} (\mathbf{x_t} - \mathbf{C_i})^\top \mathbf{W}_{xx}^{-1} (\mathbf{x_t} - \mathbf{C_i}) \tag{3.40}$$

$$\text{mit } \mathbf{W}_{xx}^{-1} : \text{inverse Gewichtungsmatrix}$$

Wählt man als Gewichtungsmatrix $\mathbf{W}_{xx}$ die diagonale Kovarianzmatrix von $\mathbf{X}$ (Varianzen), so erhält man die normierte euklidische Distanz. Bei der Berücksichtigung der vollen Kovarianzmatrix und damit der Korrelationen zwischen den einzelnen Dimensionen von $\mathbf{X}$, bezeichnet man Gl. (3.40) als Mahalonobis-Distanz. Für die Verwendung der Mahalonobis-Distanz muss $\mathbf{W}_{xx}$ invertierbar sein. Dieses Distanzmaß wird auch in den Simulationen verwendet. Im Gegensatz zu den Kovarianzen des GMM-Verfahrens, die für jede Komponente i einzeln berechnet werden (lokale Kovarianz), wird hier nur eine große Kovarianz für alle Codevektoren einer Klasse verwendet.

Da der Datenvektor $\mathbf{x_t}$ auf den naheliegendsten Codevektor $\mathbf{C_i}$ quantisiert wird, interessiert nur die Distanz zu diesem Codevektor.

$$d_{\min}(\mathbf{x_t}) = \min_i d(\mathbf{x_t}, \mathbf{C_i}) \tag{3.41}$$

Die mittlere Distanz für die gesamte Observationsmatrix $\mathbf{X}$ ergibt dann:

$$D(\mathbf{X}) = \frac{1}{T} \sum_{t=1}^{T} d_{\min}(\mathbf{x_t}) \tag{3.42}$$

Genau genommen müsste man hier den Erwartungswert des Distanzmaßes bilden. Unter Annahme der Ergodizität [Hän97] der Trainingsdaten (siehe auch Abschnitt GMM) kann das zeitliche Mittel zur Approximation des Erwartungswertes verwendet werden.

Ein Algorithmus zur Codebuchoptimierung wurde von Linde, Buzo und Gray [LBG80] vorgeschlagen.

3.3.1 LBG-Algorithmus

Der LBG-Algorithmus von ***Linde***, ***Buzo***, ***Gray*** findet durch ein iteratives Verfahren aus abwechselnder Distanzberechnung und Verschiebung der Codevektoren ein (im Bezug auf die gewichtete euklidische Distanz) optimales Codebuch. Ein Codevektor wird dabei in jeder Iteration auf den Schwerpunkt der ihm zugeordneten Datenpunkte verschoben. Die Wahl des Startcodebuchs kann dabei für den Ausgang des Verfahrens von Bedeutung sein, wie in der unten stehenden Grafik veranschaulicht ist:

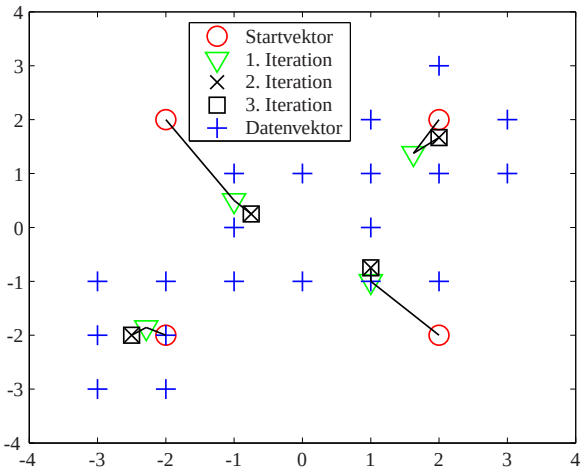


Abbildung 3.13: VQ, Start-Codebuch 1, LBG-Algorithmus

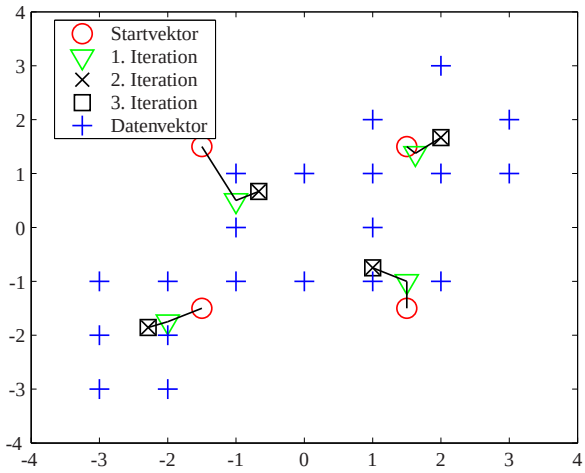


Abbildung 3.14: VQ, Start-Codebuch 2, LBG-Algorithmus

Folgende Schritte sind für die Vektorquantisierung nach [LBG80] notwendig:

1. Wahl eines Start-Codebuchs C_{init}, bestehend aus M Zufallssvektoren C_i der Dimension N
2. Quantisierung einer Observation $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_T}]$ und Berechnung der Gesamtverzerrung durch die mittlere Distanz mit Gl. (3.42)
3. Ersetzen der alten Codevektoren C_i durch das geometrische Mittel der jeweils zugeordneten Observationsvektoren x_t.

4. Wiederholung von Schritt 2, falls die Änderung der Gesamtverzerrung D gegenüber der vorherigen Iteration einen Konvergenzschwellwert unterschreitet oder die maximale Anzahl zuvor festgelegter Iterationen erreicht ist.

3.3.2 Klassifikation

Für die Klassifikation einer Testsequenz gibt es bei der Vektorquantisierung zwei Verfahren. Im ersten Verfahren wird, wie bei der Modellberechnung, die Gesamtverzerrung der Observation im Bezug auf die Codebücher aller Klassen berechnet. Die Testsequenz wird dann anschließend der Klasse $\hat{S}$ zugeordnet, welche die minimale Gesamtverzerrung ergibt. Bezeichnet man mit D_k aus Gl. (3.42) die Gesamtverzerrung einer Observation im Bezug auf eine Klasse $k = 1 \ldots S$, so erhält man mit

$$\hat{S} = \arg \min_{1 \leq k \leq S} D_k(\mathbf{X}) \tag{3.43}$$

die wahrscheinlichste Klasse für die Testsequenz.

Eine andere Möglichkeit der Klassifikation besteht in der Modifikation der Distanzberechnung aus Gl. 3.41. Hier wird zunächst die minimale Distanz einer Einzelobservation $\mathbf{x_t}$ zu den Codevektoren *einer* Klasse k bestimmt. Diese Berechnung wird für alle S Klassen durchgeführt. Anschließend erfolgt die Bestimmung der Klasse für die Einzelobservation $\mathbf{x_t}$ durch

$$\hat{S}_t(\mathbf{x_t}) = \arg \min_{1 \leq k \leq S} d_{\min k}(\mathbf{x_t}) \tag{3.44}$$

Die Testsequenz wird anschließend der Klasse zugeordnet, die am häufigsten für die gesamte Observation $\mathbf{X}$ auftrat:

$$\hat{S} = \max_{1 \leq k \leq S} H(\hat{S}_t(\mathbf{x_t}) = k) \quad \text{mit } H : \text{absolute Häufigkeit} \tag{3.45}$$

Kapitel 4

Das Simulationssystem

Im Rahmen dieser Arbeit wurde ein Simulationssystem für verschiedene Klassifikationsverfahren unter Matlab entwickelt, dessen Ablauf und Struktur im folgenden Abschnitt beschrieben wird.
Die Signalklassifikation lässt sich wie aus Abb. 4.2 ersichtlich, prinzipiell in drei voneinander weitgehend unabhängige Aufgaben einteilen:

- der Merkmalextraktion,
- der Klassen- oder Modellberechnung und
- der Klassifikation bzw. Identifikation.

Merkmalextraktion In der Merkmalextraktion werden, wie bereits in Kapitel 2 beschrieben, Eigenschaften eines Audiosignals extrahiert. Diese können z.B. spektrale oder cepstrale Informationen sowie weiterverarbeitete und reduzierte Daten (Grundfrequenz, etc.) des Signals sein. Im Simulationssystem verläuft diese Merkmalextraktion vollkommen unabhängig von den gewählten Klassifikationsverfahren, so dass zu jedem Audio-Signal zuvor Merkmaldaten extrahiert und abgespeichert werden können. Eine Ausnahme bilden hier die Audio Spectrum Basis und die Audio Spectrum Projection, welche sich immer auf eine Signal*klasse* beziehen und daher erst unmittelbar vor der Modellberechnung und der Klassifikation berechnet werden können (siehe Abschnitt 2.2).

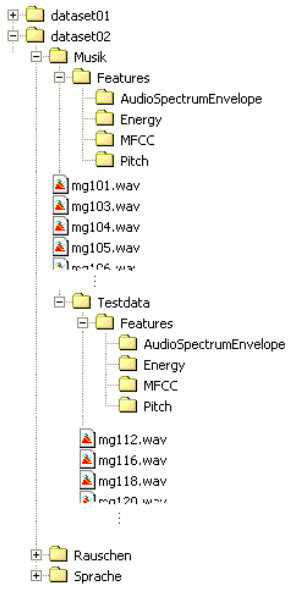


Abbildung 4.1: Simulationsstruktur

Für die nachgeschalteten Aufgabenblöcke ist eine Kenntnis der Audiodaten nicht mehr erforderlich. In einem Simulationssystem ist es sinnvoll, eine geordnete Verzeichnisstruktur aufzubauen, die den Zugriff auf die Daten für den

Anwender transparent und jederzeit reproduzierbar macht. Dazu wird für jede Signalklasse ein Verzeichnis mit den Klassennamen erstellt, in dem die Audiodaten in Form von *.wav-Dateien liegen (siehe Abb 4.1). In einem Unterverzeichnis „Features“ befinden sich weitere Ordner, die den Namen der extrahierten Merkmale tragen. Hierin werden die extrahierte Merkmalvektoren der Trainingsdaten in Form von Matlab *.mat-Dateien abgespeichert. Ein weiteres Verzeichnis im Ordner der Signalklassen heißt „Testdata“. Hierin befinden sich die Audio-Dateien der Testdaten und eine entsprechende Verzeichnisstruktur für die Merkmalvektoren dieser Testdaten.
Die manuelle Sortierung der Testdaten in die Verzeichnisse der zugehörigen Klassen ist sinnvoll, da sie die Auswertung vereinfacht. Alternativ könnte die Klassenzugehörigkeit auch im Dateinamen oder in einer externen Datenbank gespeichert sein. Diese Informationen dürfen selbstverständlich der Klassifikationsfunktion nicht bekannt sein, sondern sie dienen lediglich der Bestimmung der Trefferquote *nach* der Klassifikation.
Ein weiterer Punkt, der beachtet werden muss, ist, dass Test- und Trainingsdaten verschieden sind. Für praktisch alle Anwendungen soll die Klassifikation auf Grundlage statistischer Ähnlichkeit und nicht aufgrund der Identität von Test- und Trainingsdaten erfolgen.

Modellberechnung Für die Berechnung des Klassenmodells können prinzipiell mehrere unterschiedliche Merkmaldaten zu einer mehrdimensionalen Observation $\mathbf{X} \in \mathbb{R}^{(N \times T)}$ kombiniert werden, wobei T die Anzahl der zeitlich nacheinander erfolgten Einzelobservationen angibt (T=Zeitdimension) und N die Summe der Dimensionen in Parameterrichtung (z.B. Anzahl der Frequenzkoeffizienten plus Anzahl der Cepstralkoeffizienten plus Anzahl Pitch-Koeffizienten) ist (N=Parameterdimension). Die Kombination von mehreren (mehrdimensionalen) Merkmalen wurde in den nachfolgenden Simulationen jedoch nicht durchgeführt, um unnötige Redundanz der ohnehin rechenaufwändigen Algorithmen zu vermeiden.
Der Ursprung der Observationen ist für den Trainingsalgorithmus belanglos. Die Observationen müssen nur die Bedingung erfüllen, eine für die Signalklasse typische Verteilung zu haben, mit einer möglichst geringen Intra-Klassenvariation. Nach der Modellberechnung wird für jede Klasse eine *.mat-Datei mit den Modellparametern im jeweiligen Verzeichnis abgespeichert. Die Modellparameter haben einen geringen Datenumfang, so dass die Speicherung, Archivierung oder Übertragung von Modellparametern unproblematisch ist.

Klassifikation Die Modelle können anschließend vom Klassifikator eingelesen werden. Außerdem werden die Merkmalvektoren der betrachteten Testdatei eingelesen und zu einer Observation zusammengefasst. Es muss dabei dieselbe Kombination und Parametereinstellung der Merkmale verwendet werden, auf die auch die Modellberechnung basiert. Um dieses zu gewährleisten, werden die verwendeten Merkmale zu Merkmalgruppen (Sets) zusammengefasst und durch ein Dateinamen-Prefix (z.B. Set01) gekennzeichnet. Am Ende einer Klassifikation, die alle Testdaten aller Signalklassen nacheinander klassifiziert, erfolgt die Ausgabe der Likelyhood-Werte bzw. Distanzen in eine Log-Datei. Anhand des (bei einer Simulation vorhandenen) apriori-Wissens der Klassenzugehörigkeit einer Testdatei, kann eine statistische Auswertung der Trefferquoten erfolgen. Diese wird als Matlab-Figure und als *.mat-Datei gespeichert.

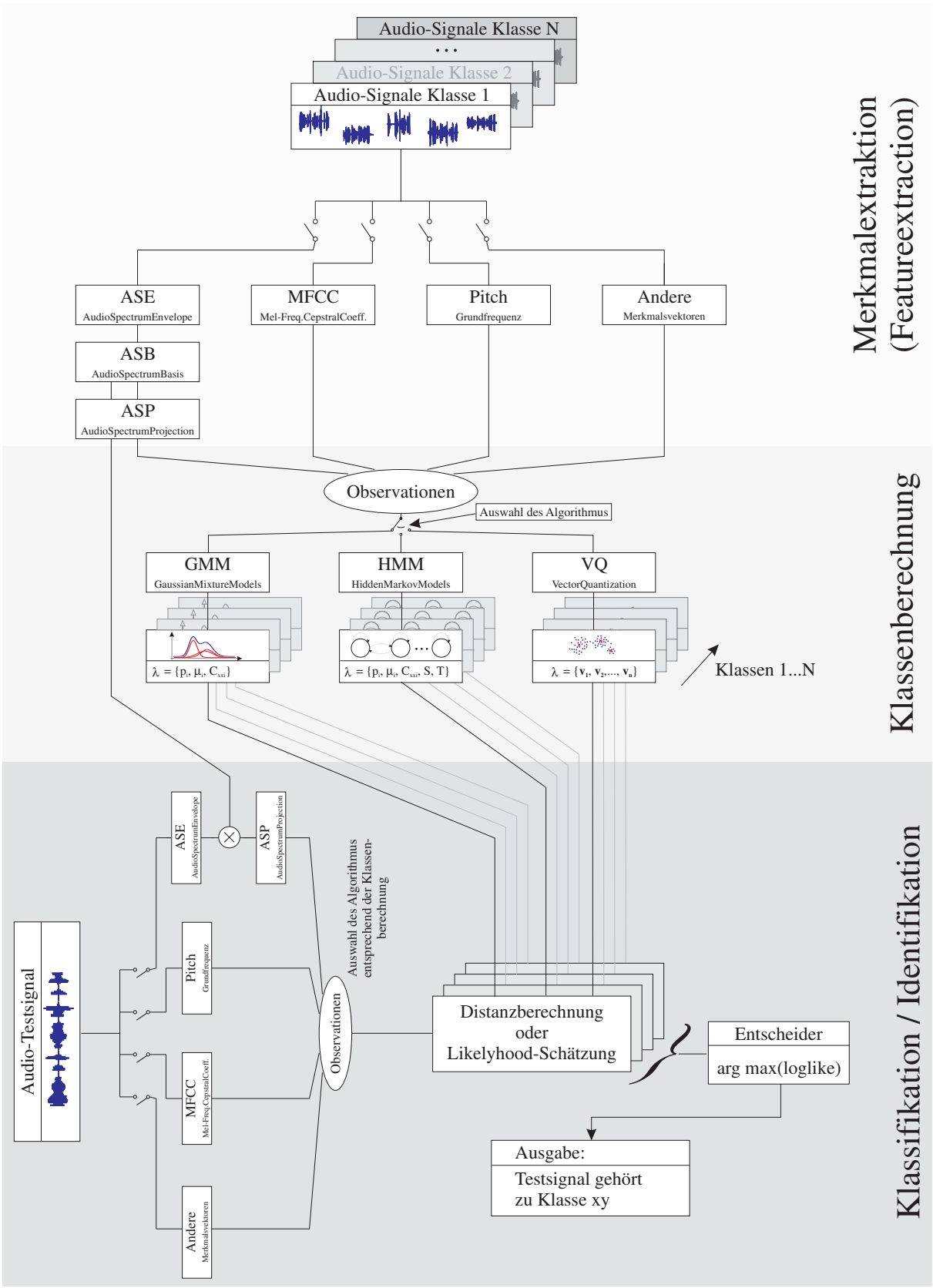


Abbildung 4.2: Simulationsstruktur

Algorithm: GMM
Feature: AudioSpectrumProjection
Sampling Frequenz: 16000
HopSize: PT10N1000F
Hi Edge: 8000
Low Edge: 62.50
Octave Resolution: 1/8
Iterations: 50
Basis Functions: 12
Means: 12
Covariance Type: diagonal

File	Time	AngelaMerkel	ChristianWulff	EberhardPiltz	EdmundStoiber	EvaHerman	FriedrichMerz	GerhardSchroeder	GuidoWesterwelle	GuntherTiersch	Harald
Set01_EvaHerman (M)_01(2)_001_ASE.mat	3.2	8867	8380	8435	8436	9237	9019	8646	8557	8450	8816
Set01_EvaHerman (M)_01(3)_001_ASE.mat	3.2	7210	6767	6842	6894	7440	7439	7063	7064	6973	7159
Set01_EvaHerman (M)_05(1)_001_ASE.mat	3.2	7851	6856	7038	7388	8876	8326	7786	7770	7704	8064
Set01_EvaHerman (M)_05(2)_001_ASE.mat	3.2	8060	7306	7448	7612	9140	8401	8029	7972	7777	8270
Set01_EvaHerman (M)_07(3)_001_ASE.mat	2.9	3223	3194	3248	3206	3676	3435	3285	3111	3249	3309
Set01_EvaHerman (M)_08(2)_001_ASE.mat	3.2	8278	7889	8138	8078	9203	8533	8332	8151	8192	8510
Set01_EvaHerman (M)_10(3)_001_ASE.mat	3.2	8366	7816	7702	7755	9035	8494	8221	8068	8046	8439
Set01_EvaHerman (M)_11(1)_001_ASE.mat	3.2	8162	7818	7911	7878	8910	8293	8158	7905	8052	8161

Abbildung 4.3: Log-Datei

Log-Datei Um die Gründe für eine Fehlklassifikation zu überprüfen, wurde für jede Klasse eine Log-Datei erstellt. Hier konnte zusätzlich, zu der „harten“ Zuordnung eines Testsignals zu einer Klasse, kontrolliert werden, welche Klasse den zweit- oder dritthöchsten Log-Likelyhood Wert hatte. Diese Informationen könnte man beispielsweise ausnutzen, um mit einem Ausschlußverfahren noch eine höhere Trefferquote zu erzielen. Die zugeordneten Audio-Dateien sind hier zur Kontrolle über Querverweise erreichbar.

Kapitel 5

Auswertungen

Im folgenden Abschnitt werden die Simulationsergebnisse beschrieben. Es wurden im Rahmen der Arbeit vier verschiedene Anwendungsgebiete der Klassifikation getestet.

- männlich-/weiblich-Klassifikation von Sprecherstimmen
- Sprecherklassifikation
- Musik-/Sprache-/Rauschen-Klassifikation
- Detektion von Musik mit Gesang und Instrumentalmusik verschiedener Stilrichtungen

Für jedes Anwendungsgebiet wurden Simulationen mit den in Kapitel 3 beschriebenen Algorithmen durchgeführt und die Klassifikation im Bezug auf die Verwendung unterschiedlicher Merkmale und deren Parameter-Einstellungen untersucht. Bei der Klassenidentifikation unterscheidet man zwischen „closed-set“ und „open-set“ Verfahren. Bei der „closed-set“ Identifikation wird jedes Testsignal genau einer Klasse zugeordnet, während beim „open-set“ auch die Zuordnung zu *keiner* Klasse, also eine Nichtzuordnung, möglich ist. Die Auswertungen werden in dieser Arbeit mit einer „closed-set“ Identifikation durchgeführt, wodurch sich die Trefferquote auf folgende Weise bestimmen lässt:

$$\text{Trefferquote in \%} \quad = \quad \frac{\text{Anzahl richtig klassifizierter Audio-Segmente}}{\text{Anzahl der Audio-Segmente einer Klasse}} \cdot 100. \qquad (5.1)$$

Die „closed-set“-Auswertung hat die Eigenschaft, dass die Trefferquote bei Verringerung der Klassenanzahl nur steigen kann. Bei einer zwei-Klassen-Entscheidung entspricht eine Durchschnittsquote von 50% bereits einer Zufallsentscheidung.
Um die Klassifikationsverfahren miteinander vergleichbar zu machen, müssen nicht nur die Parametereinstellungen der verwendeten Merkmale übereinstimmen, sondern auch ihre Modell-Ordnung M. Für das GMM ist dies die Anzahl der Komponentenverteilungen, bei dem HMM die Anzahl der Zustände und für den VQ ist die Modell-Ordnung durch die Codebuchgröße, d.h die Anzahl der Codevektoren bestimmt.

5.1 Klassifikation männlicher und weiblicher Sprecherstimmen

Für die erste Klassifikation wurde eine Datenbank aus männlichen und weiblichen Sprecherstimmen verwendet. Die Audiodaten stammen aus der Sprecherdatenbank eines MPEG-7 Software-Pakets. Als Trainingsdaten wurden jeweils 70% der Audiodaten verwendet. Dies entsprach 5,5 Minuten weiblicher Sprecherstimmen und 8,5 Minuten männlicher Stimmen. Die restlichen Daten standen für die Identifikation zur Verfügung und hatten jeweils eine Testdatenlänge von ca. 3 Sekunden. Aus den Audiodaten, die mit einer Abtastrate von $f_s = 16\,\mathrm{kHz}$ vorlagen, wurden die Audio Spectrum Envelope Merkmale mit folgenden Parametereinstellungen extrahiert:

```
hopSize = 160            % Samples, 10 ms Schrittweite, 30 ms Bloecke
hiEdge  = 8000           % Hz
loEdge  = 324.2099       % Hz
octaveResolution = '1/8' % 8 Koeffizienten pro Oktave
```

Mit den oben angegebenen Einstellungen wurden 8 Koeffizienten pro Oktave bestimmt. In den Bereich zwischen `loEdge` und `hiEdge` fielen demnach

$$\log_2(8000) - \log_2(324.2099) \cdot 8 \;\; = \;\; 37 \tag{5.2}$$

Koeffizienten. Hinzu kamen jeweils ein Koeffizient für die Energie oberhalb bzw. unterhalb der Bandgrenzen, wodurch das ASE-Merkmal letztendlich eine Dimension in Frequenzrichtung von $N = 39$ Koeffizienten hatte. Die Anzahl der Basisvektoren des ASB-Merkmals wurden entsprechend der Angaben in Tabelle 5.1 gewählt, so dass die Dimension der als Observationen verwendeten Audio Spectrum Projection-Daten entsprechend reduziert war. Die Modellberechnung und anschließende Klassifikation wurden mit den drei Verfahren GMM, HMM und VQ durchgeführt, wobei für die GMM einmal volle Kovarianzmatrizen (im folgenden als GMM/F für „full“ abgekürzt) und einmal diagonale Kovarianzmatrizen (Varianzen) verwendet wurden (kurz: GMM/D).

Die Merkmaldimension ist für die Audio Spectrum Projection (ASP)-Merkmale frei wählbar über die Anzahl der Basisvektoren (ASB) (siehe Kapitel 2.2). Für eine ASP-Dimension von 16 sowie einer ebenso großen Modell-Ordnung ergaben sich mit allen Verfahren bereits hohe Trefferquoten von über 95%. Mit der Steigerung beider Parameter auf 18 konnte mit dem HMM- und dem GMM-Verfahren mit vollen Kovarianzmatrizen sogar eine 100%ige Trefferquote erzielt werden.

Merkmal: Audio Spectrum Projection (ASP) Merkmaldimension: 16 Modell-Ordnung: 16			
Algorithmus	Trefferquote in %		
	weiblich	männlich	gesamt
GMM/D.	98.2	96.3	97.3
HMM	98.2	95.4	96.8
VQ	96.4	95.4	95.9
GMM/F	100	98.4	99.2

Merkmal: Audio Spectrum Projection (ASP) Merkmaldimension: 18 Modell-Ordnung: 18			
Algorithmus	Trefferquote in %		
	weiblich	männlich	gesamt
GMM/D.	100	96.3	98.1
HMM	100	100	100
VQ	100	95.4	97.7
GMM/F	100	100	100

Tabelle 5.1: Auswertungen männlich/weiblich Klassifikation

Weitere Klassifikationen der Testdaten wurden für die MFCC-Merkmale durchgeführt und erzielten für alle Klassifikationsverfahren Trefferquoten von 100%:

Merkmal: Mel Frequency Cepstral Coefficients (MFCC) Merkmaldimension: 24 Modell-Ordnung: 18			
Algorithmus	Trefferquote in %		
	weiblich	männlich	gesamt
GMM/D.	100	100	100
HMM	100	100	100
VQ	100	100	100

Tabelle 5.2: Auswertungen männlich/weiblich Klassifikation

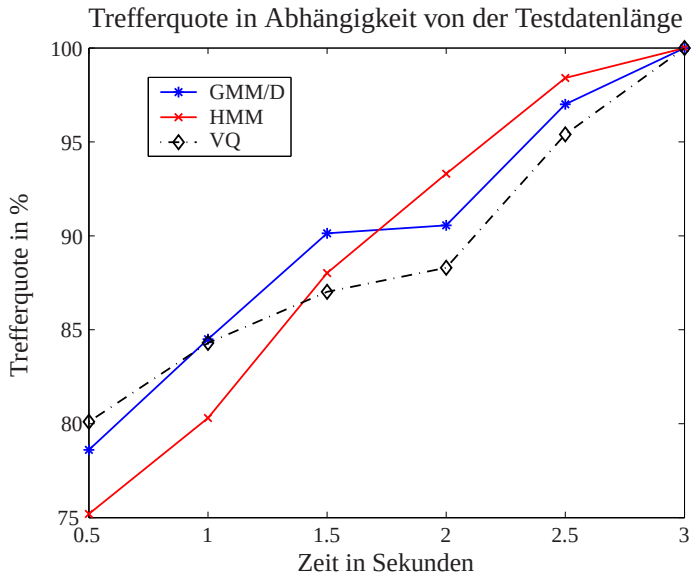


Abbildung 5.1: Klassifikation mit unterschiedlicher Testdatenlänge

Das zweite, aufwändigere GMM/F-Verfahren wurde hier weggelassen, da bereits das GMM/D eine korrekte Klassifikation ermöglichte. Für Testsignale unter 3 Sekunden sank die Trefferquote bei allen Verfahren ab (siehe Abb. 5.1), da hier nicht mehr genügend statistische Daten zur Verfügung standen. Beispiel: Für eine Schrittweite von 256 Samples sind dies bei einer Abtastfrequenz von 16 kHz nur rund 60 Merkmalvektoren pro Sekunde.

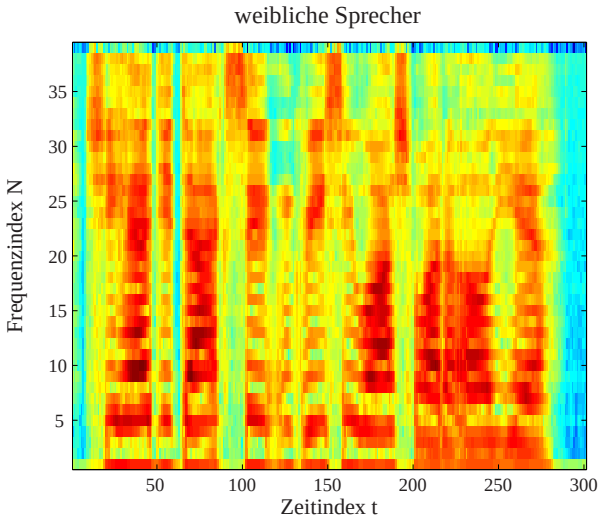


Abbildung 5.2: Audio Spectrum Envelope (ASE)

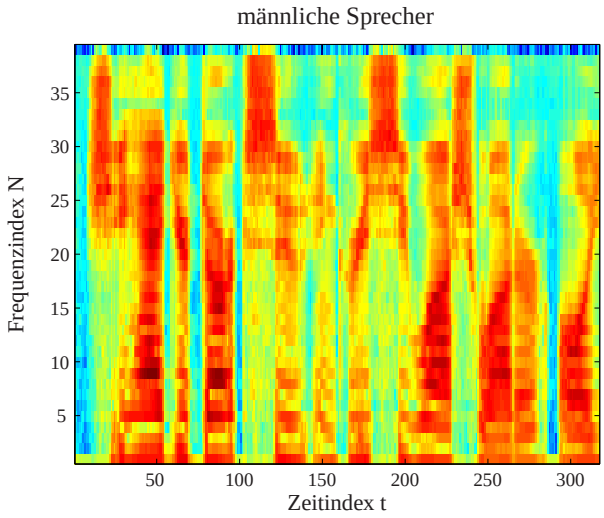


Abbildung 5.3: Audio Spectrum Envelope (ASE)

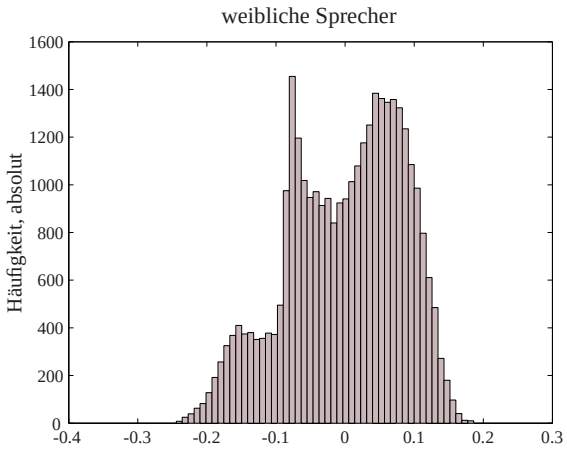


Abbildung 5.4: Histogramm eines einzelnen ASP-Vektors der weiblichen Sprecherstimmen

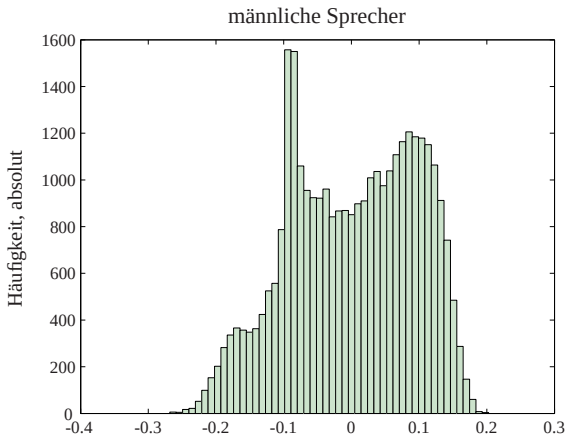


Abbildung 5.5: Histogramm eines einzelnen ASP-Vektors der männlichen Sprecherstimmen

Die hohen Trefferquoten lassen den Schluss zu, dass die statistischen Daten der beiden Klassen sehr unterschiedliche Verteilungen besitzen, die eine hohe Diskrimination zwischen den Klassen erlauben. Dennoch lassen sich in den Abbildungen 5.2 und 5.3 der Audio-Spektrogramme keine offensichtlichen Unterschiede erkennen. Die klanglichen Eigenschaften, die es dem Menschen ermöglichen, eine Klassifikation zwischen männlicher und weiblicher Stimme vorzunehmen, sind offenbar verdeckt. Auch anhand der Histogramme einzelner ASP-Vektoren, d.h. einer

einzelnen Dimension der ASP, ist kein großer Unterschied zwischen den Klassen feststellbar. Erst bei der Betrachtung mehrdimensionaler Daten liegen die Verteilungen so weit auseinander, dass eine zuverlässige Unterscheidung zwischen den Klassen möglich ist. Für diese spezielle Klassifikation lag dieser Wert bei $N = 18$ Dimensionen.

Wahl der Dimension Die richtige Wahl der Dimension ist stark von den Daten abhängig. Eine größere Dimension bewirkt prinzipiell, dass die Verteilungs-Modelle weiter auseinander liegen, da Distanzen in einem $(n + 1)$-dimensionalen orthogonalen Raum grundsätzlich größer oder gleich den Distanzen im n-dimensionalen Raum sind. Dies ist aber noch keine Garantie dafür, dass auch eine bessere Identifikation der Testdaten möglich ist. Die Voraussetzung hierfür ist, dass in den höheren Dimensionen noch Informationen über die Signalklasse enthalten sind und nicht die statistischen Daten unhörbarer oder uncharakteristischer (Stör-)Signale. Gerade bei den ASP-Merkmalen liegen in den höheren Dimensionen Daten, die keine wesentliche Information mehr über die Signalklasse enthalten. Dies liegt an der Betrags-Sortierung der Eigenvektoren durch die SVD. Auch die MFCC-Merkmale repräsentieren mit den „höheren“ Koeffizienten weniger relevante Merkmale des Signalspektrums. Eine Dimensionsreduktion, wie sie bei den ASP-Merkmalen vorgenommen wird, kann daher, neben einer bedeutenden Reduktion des Rechenaufwands, zu einer besseren Darstellung des Klassenmodells führen.
In der Statistik äußert sich der geringe Informationsgehalt der höheren Dimensionen der ASP in einer Gaußverteilung mit einem Mittelwert nahe Null und einer geringen Varianz, wie in Abb. 5.7 und 5.6 zu sehen ist. Hier ist zu erkennen, dass anhand des 20. Koeffizienten kaum noch eine Unterscheidung der Klassen möglich ist.

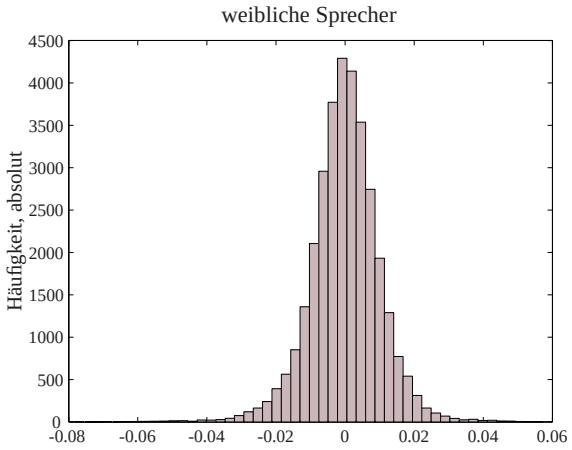


Abbildung 5.6: Histogramm der 20. ASP-Dimension

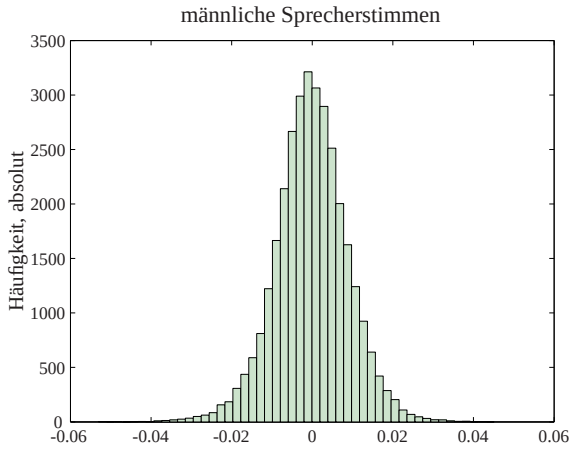


Abbildung 5.7: Histogramm der 20. ASP-Dimension

5.2 Sprecherklassifikation

5.2.1 Datenbank-Beschreibung

Die Simulationen für die Sprechererkennung basieren auf eine Gruppe aus 29 Sprechern einer nicht-kommerziellen Sprecherdatenbank. Die Aufnahmen wurden aus Videoaufzeichnungen von Nachrichten und Dokumentarsendungen extrahiert. Für jeden Sprecher existieren mehrere Audio-Dateien aus unterschiedlichen Aufnahmesituationen. Die Audiodaten variieren in der Qualität zwischen Studioaufnahmen mit hohem Signal-zu-Rauschverhältnis und Außenaufnahmen mit einer starken Geräuschkulisse. Dieses entspricht weitgehend realen Bedingungen für mögliche Anwendungen der automatischen Sprechererkennung. Als Trainingsdaten standen von jedem Sprecher mindestens 104 Sekunden Sprachaufnahmen zur Verfügung. Aus diesen Audiodaten wurden die ASE-Merkmalvektoren extrahiert. Dabei wurde eine Tiefpassfilterung auf 8 kHz durchgeführt und eine Abtastrate von 16 kHz verwendet. Die weiteren Einstellungen für die ASE Extraktion entsprachen denen aus der männlich/weiblich Klassifikation in Abschnitt 5.1.

Diese Einstellungen resultierten in eine logarithmische Spektrogrammschätzung mit $N = 39$ logarithmischen Frequenzkoeffizienten für jeden der T Zeitrahmen von den Audio-Trainingsdaten einer Sprecherklasse. Die Frequenzdimension N war bei allen Klassen konstant, die Anzahl der Zeitrahmen T (Zeitdimension) variierte in Abhängigkeit der Trainingsdaten und betrug mindestens 10400 Blöcke (=Einzelobservationen).

Die ASE-Merkmale wurden anschließend mit den klassenabhängigen ASB-Vektoren in die ASP-Merkmale umgewandelt. Um die Auswirkung der Reduktion auf die Klassifikation zu untersuchen, wurde die Anzahl der Basisvektoren schrittweise von $K = 6$ bis $K = 22$ erhöht und jeweils eine vollständige Modellberechnung und Klassifikation für die betrachteten Algorithmen durchgeführt.
Als Testdaten standen für jeden Sprecher mindestens 44 Sekunden Audio-Material aus unterschiedlichen Aufnahmen zur Verfügung, von denen, mit den gleichen Parametereinstellungen, wie bei den Trainingsdaten, ASE-Merkmale extrahiert wurden. Auch die Anzahl der Basisvektoren wurde entsprechend der Modellberechnung variiert.
Für die Identifikation wurden die ASE-Testdaten in Abschnitten zu 3 Sekunden geteilt. Die Observationsdaten mussten für jede Klasse, gegen welche die Testdaten geprüft wurden, mit Hilfe der klassenspezifischen Basisvektoren aus den ASE-Merkmalen neu berechnet werden (Details zur hierzu siehe Kapitel 2.2).
Als Referenz zu den ASP-Merkmalen wurden zusätzlich Klassifikationen mit MFCC-Merkmalen durchgeführt. Hierfür wurden 24 Koeffizienten berechnet. Bei der Extraktion der MFCC wurden folgende Einstellungen zugrunde gelegt:

```
fftLen  = 2048;  % FFT-Laenge
dist    = 512;   % Schrittweite
fs      = 16000; % Samplingfrequenz
```

5.2.2 Art der Auswertung

Die Qualität eines Klassenmodells wurde anhand einer Trefferquotenauswertung gemäß Gl. (5.1) für jede Sprecherklasse ermittelt. Da die Trefferquoten für einzelne Sprecher teilweise stark in Abhängigkeit der Parametereinstellungen für die Modellberechnung schwankten, wurde jeweils der Mittelwert aller Trefferquoten gebildet und über den veränderten Parameter aufgetragen. Diese mittlere Trefferquote über 29 Sprecher lieferte eine Aussage, wie gut im Durchschnitt die Klassifikation über eine begrenzte Anzahl von Sprechern bei den Parametereinstellungen und verwendeten Algorithmen war.

5.2.3 Algorithmische Gegebenheiten

Für die Untersuchung der Trefferquote im Bezug auf die Anzahl der ASB-Koeffizienten wurden Simulationen mit den folgenden Algorithmen durchgeführt:

- GMM/F mit vollen Kovarianzmatrizen
- GMM/D mit diagonalen Kovarianzmatrizen,
- HMM mit diagonalen Kovarianzmatrizen und
- VQ

Modell-Ordnung Die Anzahl der Gaußverteilungen betrug beim GMM-Algorithmus `nGaussians = 12`, bei dem HMM-Algorithmus entsprach dies der Anzahl der Zustände `nStates = 12` und beim Vektorquantisierer der Codebuch-Dimension `CodeDim = 12`. Der GMM-Algorithmus mit den vollen Kovarianzmatrizen verwendete für jede der 12 Gaußverteilungen eine Kovarianzmatrix der Dimension $K \times K$ (K = Anzahl der Basisvektoren, bzw. Observationsdimension).

5.2.4 Ergebnisse

Die Simulationen haben gezeigt, dass ein Anstieg der ASP-Dimension eine höhere Trefferquote zur Folge hat. Die Abhängigkeit der Trefferquote von der Anzahl der Basisvektoren ist der Abb. 5.8 zu entnehmen. Es ergeben sich nichtlineare Kurven, deren Steigung ab der Anzahl von 16 Basisvektoren nur noch sehr gering ist. Der Informationsgehalt über eine Signalklasse ist offenbar in den höheren Dimensionen der ASP nur noch sehr gering. Eine Dimensionsreduktion ist daher, vor allem zur Verringerung des Rechenaufwands, sinnvoll. Eine Verbesserung der Trefferquote bei einer geringeren Dimension konnte hingegen nicht festgestellt werden.

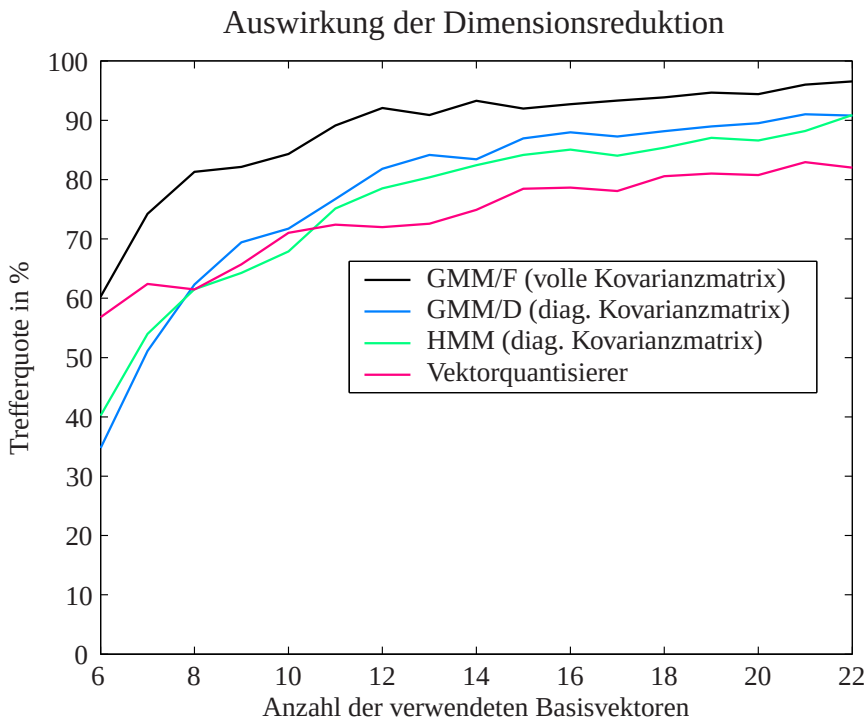


Abbildung 5.8: Auswirkung der Anzahl Basisvektoren auf die Trefferquote

Die Verwendung von vollen Kovarianzmatrizen ermöglichte eine genauere Modellierung der Verteilung als bei Diagonalmatrizen gleicher Dimension. Dieses führte zu einer höheren Trefferquote beim GMM/F mit vollen Kovarianzmatrizen, als bei den anderen Verfahren. Für das GMM/F genügte bereits eine Anzahl von 12 Basisvektoren bei einer Modell-Ordnung von $M = 12$, um eine durchschnittliche Trefferquote über 90% zu erzielen. Das HMM- so wie das GMM/D-Verfahren lagen in allen Simulationen um 6-10% unterhalb des GMM/F-Verfahrens. Die Verteilungen konnten offenbar nicht ausreichend mit der Modell-Ordnung 12 modelliert werden. Bei dem Vektorquantisierer lag die Trefferquote noch einmal 10% darunter. Durch die „harte“ Quantisierung der Merkmale auf 12 Codevektoren, ist der VQ den anderen Verfahren, die eine „weiche“ Zuordnung über Verteilungsfunktionen vornehmen, unterlegen.
Die Schwankungen der Trefferquoten einzelner Sprecher bei der schrittweisen Vergrößerung der ASP-Dimension waren teilweise recht groß. Dies ist unter anderem dadurch zu erklären, dass der Ausgang einer Modellberechnung von der Initialisierung abhängt. Im Abschnitt 3.3.1 wurde diese Eigenschaft anhand unterschiedlicher Codebuch-Initialisierungen des LBG-Algorithmus gezeigt. Prinzipiell gilt dies auch für GMM und HMM. Da mit den EM-Algorithmen nur lokale Maxima gefunden werden können, ist auch hier der Ausgang ungewiss, wenn z.B. eine zufällige Initialisierung der Parameter vorgenommen oder die Dimension der Eingangsdaten geändert wurde. In der mittleren Trefferquote über alle Sprecher wirkt sich das allerdings nicht besonders aus.
Bei manchen Sprecherklassen trat ein sprunghafter Anstieg der Trefferquote auf, wenn die Dimension der ASP um eins erhöht wurde. Eine solche Beobachtung deutete darauf hin, dass zuvor wichtige Komponenten der Sprecherklasse durch die Dimensionsreduktion unterdrückt

wurden.
Fehlklassifikationen, die sowohl von den verwendeten Merkmalen als auch von den Algorithmen unabhängig waren, wiesen auf eine hohe Intra-Klassenvariation hin. Dies trat zum Beispiel bei den Sprechern Friedrich Merz, Gerhard Schröder, Guido Westerwelle, Maybrit Illner, Peter Frey und Roland Koch auf. Ein Vergleich der Audiodaten zeigte, dass hier besonders unterschiedliche Aufnahmesituationen auftraten (z.B. gehobene, laute Stimme in einer Bundestagsrede gegenüber Interview im Studio). Wenn für derartige Signale trotz Optimierung der Modellparameter keine korrekte Identifikation möglich ist, muss überprüft werden, ob

1. die Merkmale ausreichend Information über die Klasse liefern.
2. die Trainingsdaten repräsentativ für die Signalklasse sind und damit die vorausgesetzte Ergodizität erfüllt ist.

Auf den folgenden Seiten sind die Trefferquoten der jeweils besten Klassifikationen der vier Verfahren abgebildet. Beim GMM/F-Verfahren wurden 24 der 29 Sprecher zu 100% erkannt nur 2 Sprecher lagen unterhalb der 90%-Marke. Wie bereits im Kapitel 3 angenommen wurde, kann das HMM bei den gegebenen Eingangssignalen keine zusätzlichen Informationen aus dem zeitlichen Verlauf der Verteilung gewinnen. Dies kann man damit erklären, dass es offenbar keine sprechertypische Abfolge bestimmter spektraler Muster gibt, die durch die Gaußverteilungen modelliert werden. Eine wiederkehrende Abfolge spektraler Muster wäre hingegen beim textabhängigen Sprechertraining feststellbar. Da es sich bei den Trainingsdaten einer Klasse aber um völlig unterschiedliche Aufnahmen eines Sprechers handelt, die weder im Text noch in ihrer Länge übereinstimmen, sind sprechertypische Zustandsübergänge, von denen das HMM gegenüber den anderen Verfahren profitieren würde, kaum vorhanden. Da das hier verwendete HMM diagonale Kovarianzmatrizen hat, können die Trefferquoten des HMM im Mittel nicht höher werden, als die der GMM mit diagonalen Kovarianzmatrizen.

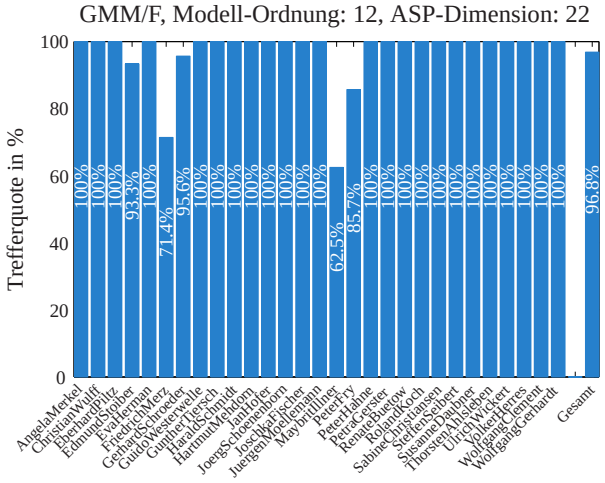


Abbildung 5.9: GMM/F

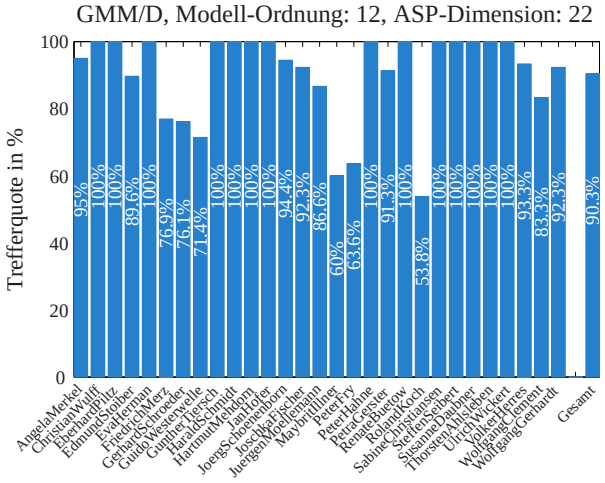


Abbildung 5.10: GMM/D

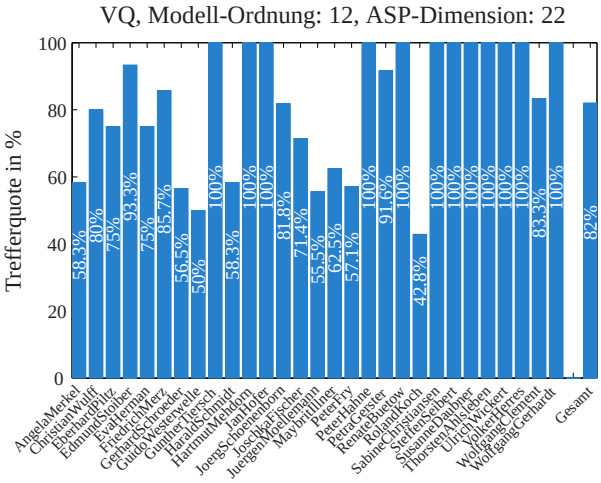


Abbildung 5.11: VQ

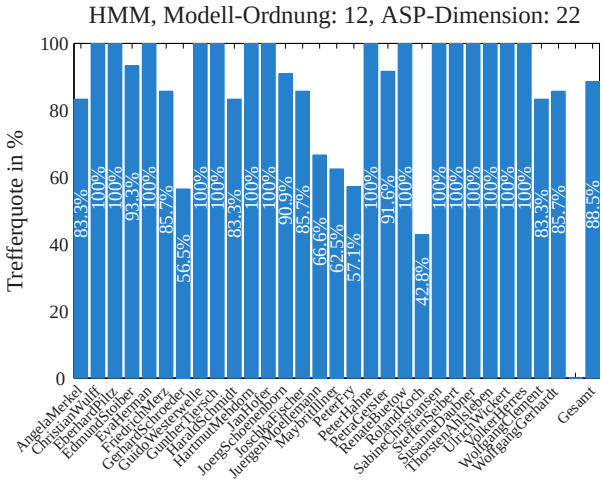


Abbildung 5.12: HMM

Als Vergleich zu den ASP-Merkmalen wurden Klassifikationen mit den MFCC-Merkmalen durchgeführt. Insgesamt ergaben sich meist geringere Trefferquoten (siehe Abb. 5.14-5.16). Zwischen den GMM/F und dem GMM/D-Verfahren ergaben sich keine Unterschiede. Das kann daran liegen, dass die MFCC-Koeffizienten weniger korreliert sind, als die ASP-Vektoren. In diesem Fall würde eine Berücksichtigung der Kovarianz keine Auswirkung haben. Die MFCC-Merkmale sind aufgrund einer sehr unterschiedlichen Varianz der einzelnen Koeffizienten problematischer für den Optimierungsalgorithmus des GMM. Hier mussten teilweise algorithmische Anpassungen gemacht werden, wie z.B. die Begrenzung auf eine minimale Varianz der Komponentenverteilungen. Nach D.A. Reynolds, der ebenfalls Begrenzungen der minimalen Varianz in [RR95] vornahm, ist dieser Parameter für jedes Daten- und Merkmal-Set in Abhängigkeit von der Modell-Ordnung empirisch zu ermitteln.

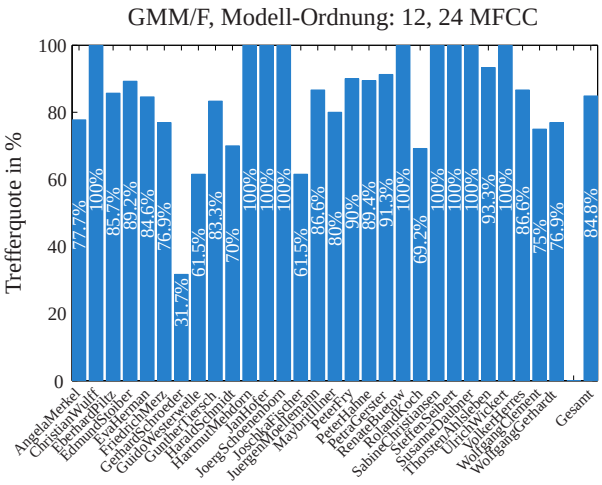


Abbildung 5.13: GMM/F

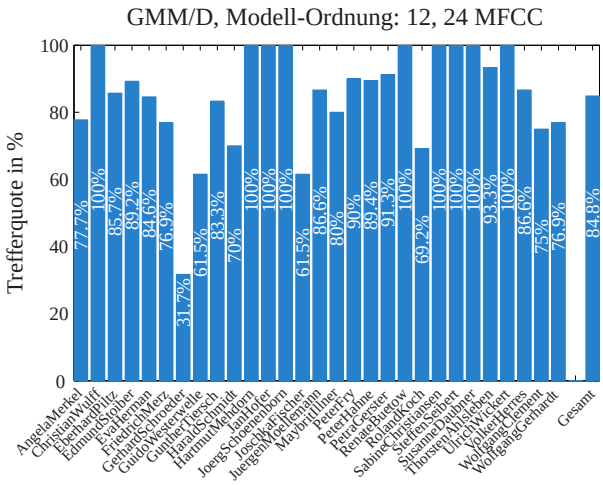


Abbildung 5.14: GMM/D

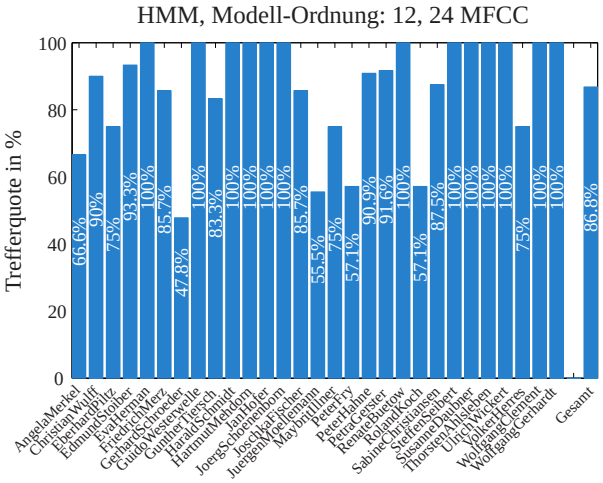


Abbildung 5.15: HMM

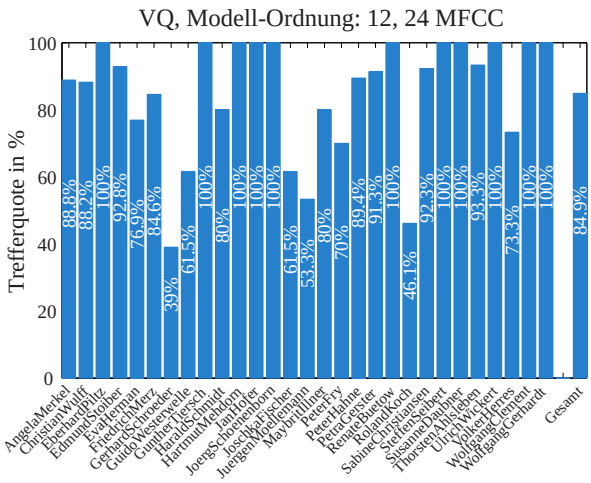


Abbildung 5.16: VQ

5.2.5 Auswirkungen der Testdatenlänge auf die Klassifikation

Für das GMM/F-Verfahren mit ASP-Merkmalen wurde zusätzlich noch eine Untersuchung der Testdatenlänge gemacht. Die Länge der Audiodaten wurde dabei von 0.5 Sekunden schrittweise bis auf 4.5 Sekunden erhöht sowie zusätzlich auf 10 und 15 Sekunden gesetzt. Mit unterschiedlichen Testsignallängen wurde jeweils eine Identifikation mit dem GMM/F-Modell aus Abb. 5.9 durchgeführt. Dabei wurde festgestellt, dass bereits 3 Sekunden Testmaterial ausreichen, um eine hohe Trefferquote von über 94% zu erzielen. Eine weitere Steigerung war mit 10 bzw. 15 Sekunden Testdatenlänge möglich. Diese ist jedoch überwiegend darauf zurückzuführen, dass hier Signale aus unterschiedlichen Aufnahmesituationen zusammengefasst wurden. Die so „gemittelten“ statistischen Daten haben dann eine größere Ähnlichkeit zu dem Modell, als die einzelnen Testsignale.

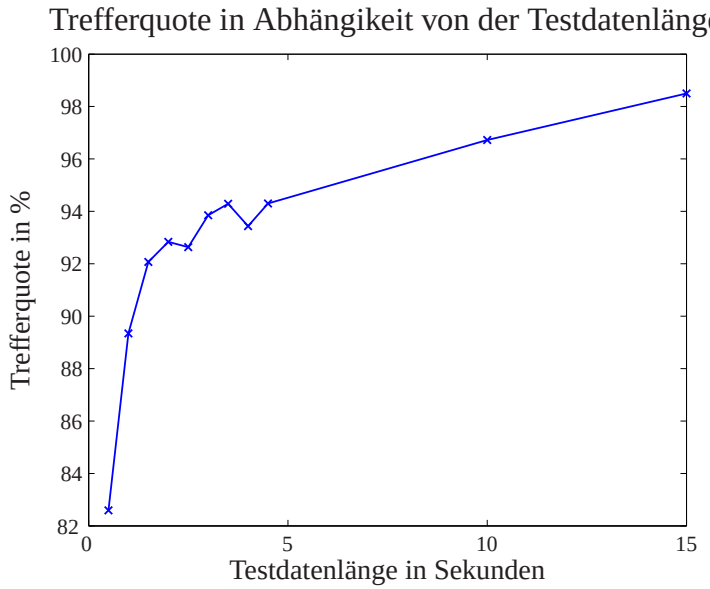


Abbildung 5.17: GMM/F Klassifikation mit unterschiedlicher Testdatenlänge

5.2.6 Untersuchung des Konvergenzverhaltens für den EM-Optimierungsalgorithmus

Zur Untersuchung der Konvergenz bei der Modelloptimierung wurde nach jeder Iteration die Differenz zwischen den Parametern Mittelwert und Varianz nach Gl. (5.3 bzw. 5.4) bestimmt und über die Anzahl der Iterationen aufgetragen. Der Verlauf und die Konvergenzgeschwindigkeit sind dabei von der Statistik der Trainingsdaten abhängig. Die Abbildungen 5.18 und 5.19 zeigen die Änderung von Varianz und Mittelwert für zwei unterschiedliche Sprecher.
Die Parameterabweichung des neuen Modells $\overline{\lambda}$ zum Modell λ aus der vorherigen Iteration wurde mit folgenden Gleichungen bestimmt:

$$\Delta\mu = \sum_{i=1}^{M} ||\overline{\boldsymbol{\mu_i}} - \boldsymbol{\mu_i}|| \tag{5.3}$$

$$\Delta C_{xx} = \sum_{i=1}^{M} ||\overline{\mathbf{C_{xxi}}} - \mathbf{C_{xxi}}|| \tag{5.4}$$

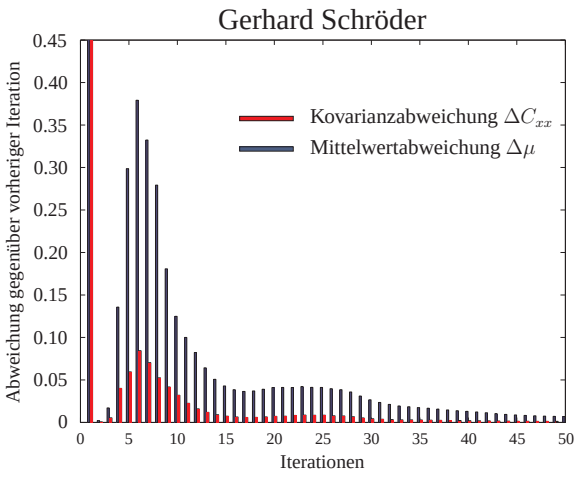


Abbildung 5.18: Konvergenzverhalten

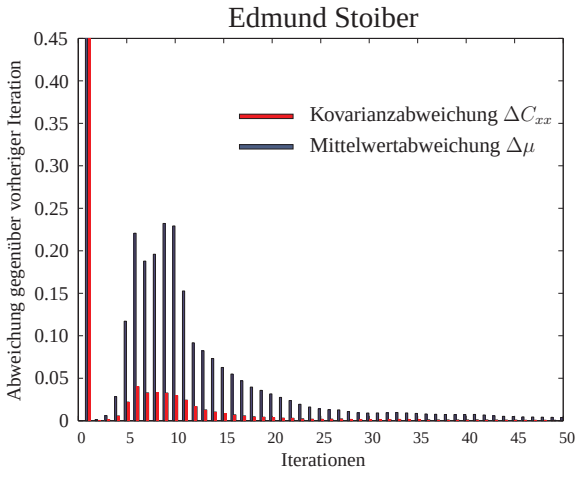


Abbildung 5.19: Konvergenzverhalten

Die Verläufe zeigen, dass die Konvergenzgeschwindigkeit von den Eingangsdaten abhängig ist. Während die Sprecherklasse in Abb. 5.19 offenbar schon nach ca. 30 Iterationen konvergiert ist, benötigt die Sprecherklasse in Abb. 5.18 ca. 45 Iterationen, um einen Konvergenzschwellwert von 0.02 zu unterschreiten. Aus den Abbildungen ist aber nicht zu ermitteln, wie sich geringe Änderungen der Modellparameter auf die Klassifikation auswirken. Daher wurden in einer weiteren Simulation die Modellparameter nach jeder fünften Iteration gespeichert, und es wurde jeweils eine Klassenidentifikation mit diesen temporären Modellen durchgeführt. Die Trefferquoten wurden dann über die Anzahl der Iterationen grafisch in Abb. 5.20 aufgetragen.

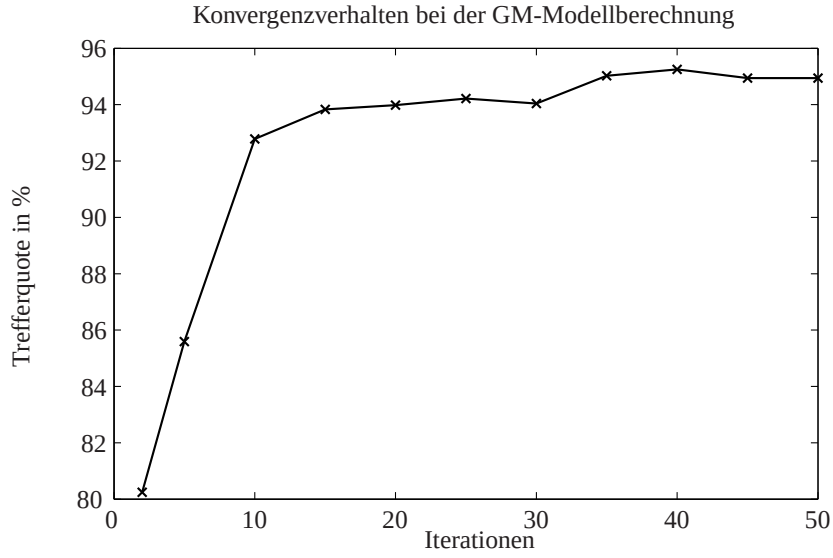


Abbildung 5.20: Konvergenzverhalten GMM

Hier bestätigt sich, dass bei den Sprecherdaten nach ca. 40 Iterationen das GMM konvergiert ist. Die Trefferquote bleibt nach weiteren Optimierungen nahezu konstant. Leichte Schwankungen bzw. Verschlechterungen der Trefferquote hingegen sind zufällig bedingt.

5.3 Musik/Sprache/Rauschen-Klassifikation

Herkömmliche Verfahren zur Unterscheidung von Musik, Sprache und Rauschen basieren auf einer analytischen Untersuchung der Signalspektren. Der Schwierigkeit bei diesen Verfahren liegt darin, geeignete Schwellwerte zu finden, um möglichst unterschiedliche Signale aus den drei Klassen zu identifizieren. Oftmals lassen sich diese Schwellwerte nicht adaptiv anpassen, womit die Definition, was als Musik, Sprache oder Rauschen bezeichnet wird, vorab festgelegt ist. Verwendet man hingegen ein statistisches Klassifikationsverfahren, kann dieses jederzeit, ohne Eingriffe in den Algorithmus, auf neue Signalklassen trainiert werden.
Die Musik/Sprache/Rauschen-Klassifikation wurde mit dem HMM und dem GMM/F-Verfahren unter Verwendung der ASP-Merkmale durchgeführt. Dazu wurden möglichst unterschiedliche Signale der drei Klassen zusammengestellt, unter Inkaufnahme einer relativ hohen Intra-Klassenvariation. Als Musik wurden Trainingsdaten verschiedener Stilrichtungen (Klassik, Rock, Pop, A-Capella Gesang, usw.) verwendet. Die Sprachsignale stammten aus unterschiedlichen Sprecherdatenbanken und enthielten sowohl männliche als auch weibliche Sprecherstimmen. Als Rauschsignale wurden Applaus, Wind, weisses Rauschen, Umwelt- und Industriegeräusche verwendet. Die Projektion der ASE-Merkmale erfolgte auf 18 Basisvektoren. Von diesen ASP-Merkmalen wurden Klassenmodelle der Ordnung $M = 18$ berechnet. Die Trefferquoten sind in Abb. 5.21 und 5.22 aufgeführt.

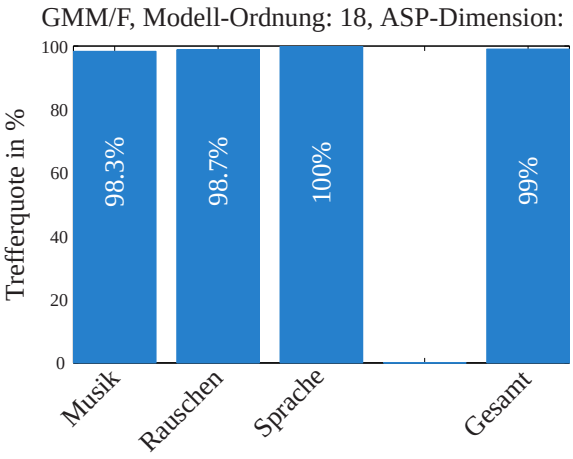


Abbildung 5.21: GMM/F

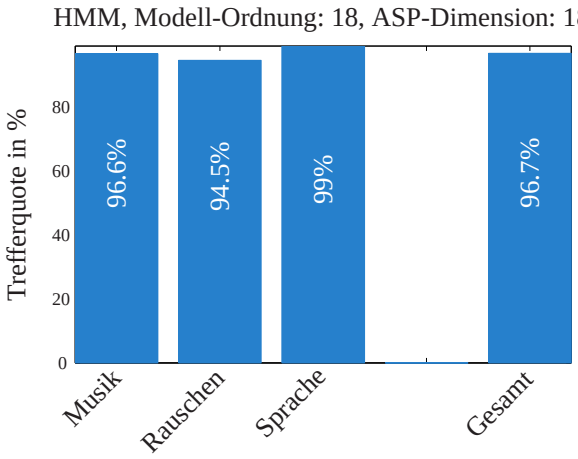


Abbildung 5.22: HMM

Das GMM mit vollen Kovarianzmatrizen erzielte auch hier die höchste Trefferquote, daher wurde für dieses Modell überprüft, wie sich die Testdatenlänge auf die Trefferquote auswirkt. Dazu wurden die Länge der Testdaten in 0.5 Sekunden Schritten von 0.5 bis 3 Sekunden variiert und die Trefferquoten ermittelt.

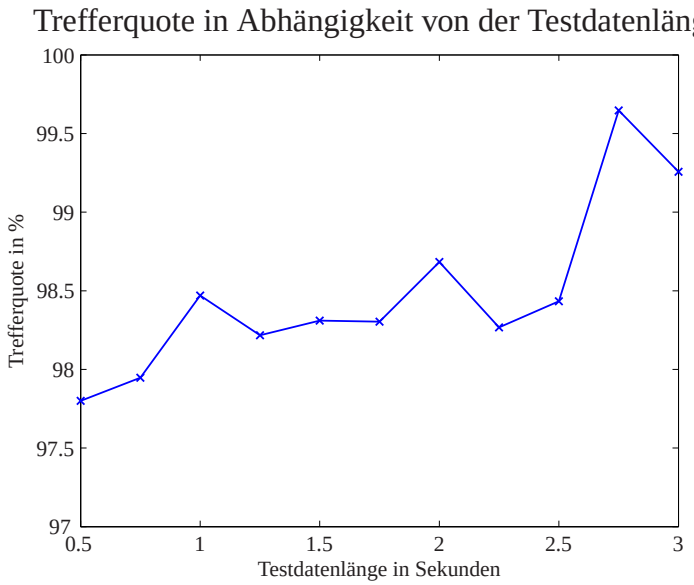


Abbildung 5.23: GMM/F Musik/Sprache/Rauschen-Klassifikation

Bereits bei einer Sekunde Testdatenlänge lag die durchschnittliche Trefferquote über 98%. Die Observationssequenz bestand hier nur noch aus 96 Datenvektoren. Da offensichtlich schon wenige statistische Daten ausreichten, um relativ sicher zwischen den drei Signalklassen zu unterscheiden, wurde für das GMM noch ein weiterer Versuch durchgeführt. Aus den Testsequenzen der drei Signalklassen wurde eine Audio-Datei von 35 Sekunden Länge zusammengeschnitten. Die Abfolge der einzelnen Signale war dabei zufällig und die Länge der Abschnitte betrug zwischen 1.5 und 5 Sekunden. Anschließend wurde eine blockweise Klassifikation des ASP-Merkmals der Audio-Datei durchgeführt und das Ergebnis über die Zeitachse aufgetragen (siehe Abb. 5.24). Die Schrittweite betrug 0.1 Sekunden und die Analyseblocklänge 1 Sekunde.

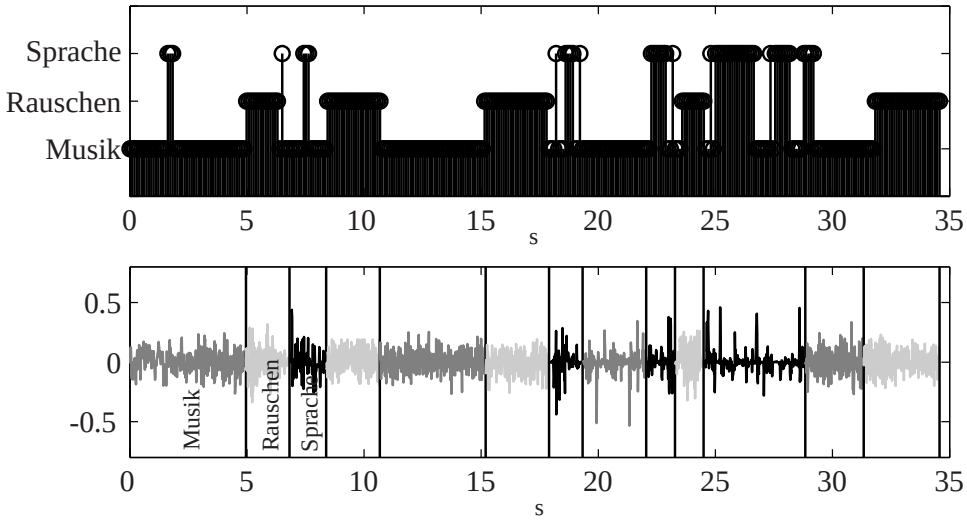


Abbildung 5.24: Musik/Sprache/Rauschen-Erkenner

Das Ergebnis der Klassifikation ist als Zustandsfolge im oberen Teil der Grafik aufgetragen. Die tatsächliche Signalklasse ist den durch unterschiedliche Grautöne gekennzeichneten Blöcken im unteren Teil zu entnehmen. Die Abschnitte mit Musik und mit Rauschen wurden fast vollständig richtig klassifiziert. Sprache wurde teilweise fälschlich als Musik klassifiziert. Die Nachteile einer blockweisen gegenüber einer ganzheitlichen Klassifikation liegen darin, dass hier gemischte Signale auftreten können. Problematisch sind dabei vor allem Klassen, die aufgrund ihrer sehr unterschiedlichen Trainingsdaten eine hohe Intra-Klassenvariation haben (wie in diesem Fall „Musik“). Ein gemischtes Testsignal mit nur geringen Anteilen der Klasse (Musik) wird dann sofort dieser zugeordnet. Zu einer Verbesserung könnten empirisch ermittelte Gewichtungsfaktoren der Klassen, die Ausnutzung von a-priori Wissen über die Auftrittswahrscheinlichkeit sowie die Festlegung einer minimalen Zustandsdauer (Zustand = Musik, Rauschen oder Sprache) sein. Diese Anpassungen sollen hier nicht weiter betrachtet werden.

5.4 Musik instrumental/Musik mit Gesang-Klassifikation

Nachdem in den vorangegangenen Simulationen die Zuordnungen der Signale zu den Klassen recht eindeutig waren, sollen nun noch statistisch sehr nahe beieinander liegende Daten betrachtet werden. Dazu wurden aus Musik unterschiedlichster Stilrichtungen Abschnitte mit Gesang und instrumentale Abschnitte extrahiert und in diese zwei Klassen aufgeteilt. Die Signale wurden mit einer etwas höheren Frequenz abgetastet, als bei den anderen Klassifikationen. Die ASE-Merkmale wurden mit den folgenden Einstellungen extrahiert:

```
fs      = 24000          % Hz
hopSize = 240            % Samples
hiEdge  = 12000          % Hz
loEdge  = 192.78         % Hz
octaveResolution = '1/8' % 8 Koeffizienten pro Oktave
```

Die Testdaten hatten jeweils eine Länge von 3 Sekunden. Die Trefferquoten lagen hier erwartungsgemäß deutlich niedriger, als bei den anderen Klassifikationen. Bei einer durchschnittlichen Erkennungsrate von 77% ist eine Unterscheidung zwischen Musik und Musik mit Gesang anhand statistischer Daten der Merkmale aber prinzipiell möglich. Die Fehlklassifikationen sind hier auch wieder auf ein Ungleichgewicht zwischen den Klassen zurückzuführen.

Merkmal: Audio Spectrum Projection (ASP) Merkmaldimension: 22 Modell-Ordnung: 22			
Algorithmus	Trefferquote in %		
	Musik instrumental	Musik mit Gesang	gesamt
GMM-diag.	68,6	85,0	76,8
HMM	35,1	96,1	65,6
VQ	83,2	42,5	62,8

Tabelle 5.3: Auswertung für verschiedene Musikrichtungen

Bei der Einschränkung der Musikrichtung auf einen Interpreten ergaben sich wesentlich bessere Ergebnisse, wie aus Tabelle 5.4 ersichtlich ist.

Merkmal: Audio Spectrum Projection (ASP) Merkmaldimension: 22 Modell-Ordnung: 22			
Algorithmus	Trefferquote in %		
	Musik instrumental	Musik mit Gesang	gesamt
GMM-diag.	85.4	97.5	91.4
HMM	35.4	100	67.7
VQ	89.5	85.3	87.4

Tabelle 5.4: Auswertung für einen Musikstil/Interpreten

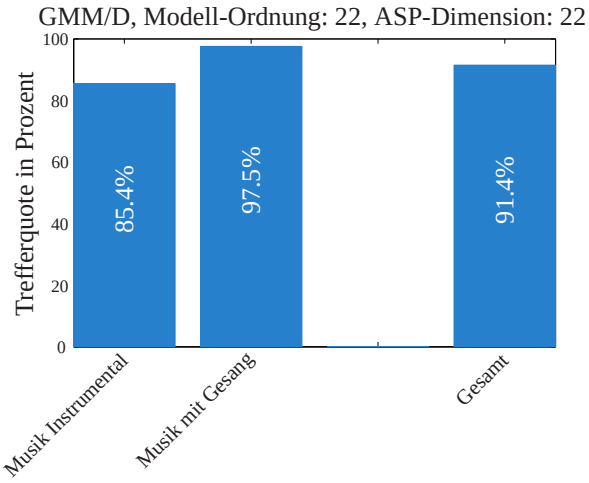


Abbildung 5.25: Musik instrumental/mit Gesang - Klassifikation

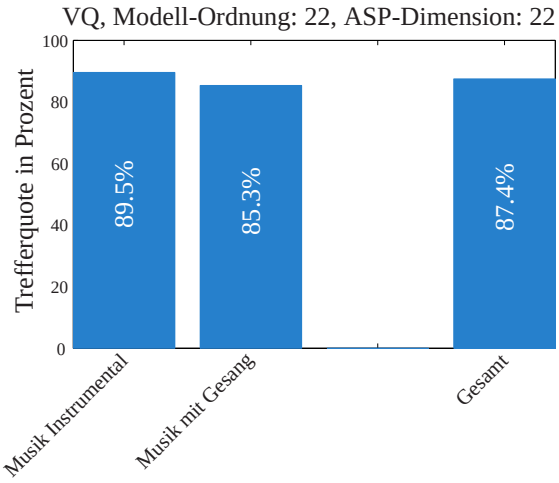


Abbildung 5.26: Musik instrumental/mit Gesang - Klassifikation

Eine Klassifikation mit dem GMM/F brachte für die gegebenen Signale keine Vorteile gegenüber dem GMM/D-Verfahren. Es kam in den Simulationen immer wieder zu Singularitäten bei

der Inversion der Kovarianzmatrizen. Begrenzungen der Werte zur Erhaltung der Stabilität führten hingegen zu einer Verschlechterung des Modells. Hier konnten noch keine geeigneten Werte gefunden werden. Ähnliche Schwierigkeiten ergaben sich für das HMM. Bei einer Trefferquote von unter 50% ist das Modell für eine „closed-set“ Identifikation als fehlerhaft anzusehen. Derartige Probleme traten nur bei sehr großen Datensätzen auf. Dies deutet auf Rundungsfehler bei der Multiplikation sehr kleiner Werte hin. Hierfür müssen noch geeignete Maßnahmen (wie z.B. Skalierung oder Datensatzbegrenzung) gefunden werden. Insgesamt konnte aber festgestellt werden, dass eine Unterscheidung zwischen instrumentaler Musik und Musik mit Gesang mit einer relativ hohen Sicherheit möglich ist, wenn zuvor eine Einschränkung der Trainingsdaten auf eine bestimmte Musikrichtung oder einen Interpreten vorgenommen wurde.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wurde die Klassifikation von Audio-Signalen mit Hilfe unterschiedlicher Verfahren untersucht, die auf eine Modellbildung stochastischer Daten basierten.

In Kapitel 2 wurden dazu sowohl klassische als auch neuartige Verfahren zur Merkmalextraktion vorgestellt. Neben den in der Sprechererkennung am häufigsten verwendeten MFCC-Merkmalen, sind hier besonders die Merkmale aus dem MPEG-7 Standard untersucht worden. Diese basieren auf eine dimensionsreduzierte Darstellung von Audio-Spektrogrammen, die im Hinblick auf die Anwendungsgebiete Klassifikation und Meta-Daten Extraktion optimiert sind. Da zu diesen Merkmalen bisher nur wenig Literatur verfügbar war, wurde besonderer Wert auf die theoretische Begründbarkeit für den erfolgreichen Einsatz in der Signalklassifikation gelegt. In den Simulationen wurde dazu unter anderem die Auswirkung der Dimensionsreduktion auf die Klassifikation untersucht. Der Vergleich zu den klassischen MFCC-Merkmalen zeigte, dass sich mit den Audio Spectrum Projection (ASP)-Merkmalen genauso gute Ergebnisse mit weniger Daten erzielen ließen, was zu einer Verringerung des Rechenaufwands führte. Diese Einsparung ermöglichte die Verwendung aufwändigerer Methoden zur Klassenmodellierung, die besonders bei geringer Dimension der Eingangsdaten eine bessere Modellbeschreibung gegenüber herkömmlichen Verfahren lieferten.

Der Schwerpunkt der Arbeit lag in der Untersuchung der drei Klassifikationsverfahren

- Gaussian Mixture Models
- Hidden Markov Models und
- Vektorquantisierung.

Die Verfahren haben gemeinsam, dass sie mit einer möglichst geringen Anzahl Parametern versuchen, die Verteilungen der stochastischen Daten einer Signalklasse modellhaft zu beschreiben. Während durch die Codevektoren eines Vektorquantisierers nur die Schwerpunkte der Daten-Cluster („Datenwolken“) erfasst werden, können bei den anderen Verfahren zusätzlich die Verteilungen um diese Mittelwerte approximiert werden. Zudem ist die Quantisierung eines Datenwerts auf einen Codevektor „fest“, während beim HMM und GMM eine „weiche“ Zuordnung der Datenwerte über Gaußverteilungen erfolgt. Gegenüber dem GMM kann das HMM

eine zeitliche Abfolge der Verteilungen berücksichtigen. In den Simulationen wurde festgestellt, dass dieser Vorteil offenbar nur dann ausgenutzt werden kann, wenn die Signale eine zeitliche Struktur besitzen. Für die in den Simulationen verwendeten Signalklassen war dies nicht der Fall.

Die für die Klassifikation verwendeten Merkmale sind in der Regel mehrdimensional oder werden aus unterschiedlichen Merkmalvektoren zusammengesetzt. Dies hat häufig zur Folge, dass statistische Abhängigkeiten zwischen den Dimensionen bestehen. Diese Abhängigkeiten werden von Verfahren, die nur die Varianzen der Merkmalvektoren berücksichtigen, nicht erfasst. Für eines der in dieser Arbeit betrachteten GMM-Verfahren wurden daher volle Kovarianzmatrizen verwendet. Dies führte zu einer verbesserten Darstellung der Klassenmodelle und resultierte in einer höheren Trefferquote, als bei den anderen Verfahren. Unter gewissen Umständen (große Datensätze, ungünstige Initialisierung der Parameter) konnte es jedoch zu Singularitäten bei der Berechnung bzw. Inversion der Kovarianzmatrix kommen. Weitere Untersuchungen, um die Stabilität der Kovarianzmatrizen zu gewährleisten, sind deshalb notwendig.

Die Literatur zur Audio-Klassifikation beschäftigt sich überwiegend mit Sprachsignalen, daher wurden auch hier Simulationen anhand einer Sprecherdatenbank durchgeführt. Die Existenz umfangreicher - aber leider größtenteils kommerzieller - Sprecherdatenbanken machen inzwischen objektive Vergleiche zwischen unterschiedlichen Klassifikationsverfahren möglich. Die Klassifikation anderer Audiodaten, wie z.B. Musik, ist hingegen ein nicht so häufig untersuchtes Gebiet. Die Verteilungen der stochastischen Merkmaldaten haben dabei eine wesentlich größere Variabilität, als die Merkmale der spektral stark begrenzten Sprachsignale. In den Simulationen wurde daher untersucht, ob auch eine Klassifikation mit anderen Audio-Signalen möglich ist und wie sich verschiedene Parameter auf die Klassifikation auswirken.

Neuartig war die Kombination der MPEG-7 Merkmale mit GMM-Klassifikationsverfahren. Die im MPEG-7 Standard definierte Klassifikation basiert auf Hidden Markov Modellen, deren höhere Komplexität keinen Gewinn bei zeitlich unstrukturierten Daten liefert. Durch gezielte Abwägung zwischen Informationsgehalt und Dimensionreduktion der Audio Spectrum Projection Merkmale in Kombination mit dem GMM, könnten in Zukunft auch Echtzeit-Klassifikationen realisiert werden. Einen Ansatz hierfür lieferte die Simulation mit einer kontinuierlichen Musik/Sprache/Rauschen-Klassifikation. Bislang unbekannt war, ob sich instrumentale Musik von Musik mit Gesang über statistische Verteilungen ihrer Merkmale unterscheiden lässt. Die Simulationen zeigten, dass eine Klassifikation möglich ist, aber die Trefferquote stark von der Definition der Signalklasse abhängt. Unter gleichzeitiger Einbeziehung vieler Musikstile, war eine Unterscheidung der Klassen wesentlich unsicherer, als bei der Klassifikation einer einzigen Musikgruppe.

Insgesamt konnte festgestellt werden, dass die Klassifikation mit allen Verfahren sowie verschiedenen Merkmaldaten möglich ist. Mit dem GMM mit vollen Kovarianzmatrizen konnten dabei die besten Ergebnisse erzielt werden. Es konnte somit gezeigt werden, dass statistische Modelle, die durch iterative Verfahren „trainiert“ werden, eine Klassifikation von Audio-Signalen für unterschiedlichste Anwendungsgebiete ermöglichen.

Anhang A

Anhang

A.1 Singulärwertzerlegung Singular Value Decomposition (SVD)

A.1.1 Allgemeine Definition

Jede reelle Matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ kann so in drei Matrizen zerlegt werden, dass folgende Gleichung erfüllt ist (Singulärwertzerlegung):

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} \tag{A.1}$$

Dabei sind $\mathbf{U}$ und $\mathbf{V}$ orthonormale Matrizen der Dimensionen $M \times M$ bzw. $N \times N$. $\mathbf{S}$ ist eine diagonale $M \times N$ Matrix, welche die Singulärwerte von $\mathbf{A}$ enthält.
Für $M = 3$ und $N = 2$ ist, ergibt sich:

$$\underset{\mathbf{A}}{\begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}} = \underset{\mathbf{U}}{\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}} \underset{\mathbf{S}}{\begin{pmatrix} s_1 & 0 \\ 0 & s_2 \\ 0 & 0 \end{pmatrix}} \underset{\mathbf{V}^{\top}}{\begin{pmatrix} \cdot & \cdot \\ \cdot & \cdot \end{pmatrix}}$$

Die Singulärwerte werden, gemäß Vereinbarung, in absteigender Reihenfolge nach ihrem Betrag sortiert, so dass $s_j \leq s_i$, für $j > i$. Es gibt exakt $l = \min(M, N)$ Singulärwerte $s_j, j = 1, \ldots, l$, von denen auch einige Null sein können. Die Anzahl r der nicht verschwindenden Singulärwerte bezeichnet man als Rang[1] der Matrix $\mathbf{A}$ (d.h. $s_j = 0, j = r + 1, \ldots, l$). Falls $r = l$ gilt, bezeichnet man $\mathbf{A}$ als Matrix mit vollem Rang. F

[1] Rang = Anzahl linear unabhängiger Spalten, bzw. Zeilen

A.1.2 Berechnung der SVD

Die Ausdrücke für die einzelnen Faktoren der SVD sind folgendermaßen zu bilden. Multipliziert man Gleichung A.1 von rechts bzw. links mit $\mathbf{A}^\top$, erhält man:

$$\mathbf{A}^\top = \mathbf{V}\mathbf{S}^\top\mathbf{U}^\top \tag{A.2}$$

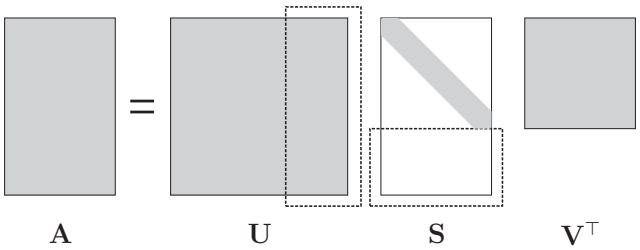
$$\Rightarrow \quad \mathbf{A}\mathbf{A}^\top = \mathbf{U}\mathbf{S}\underbrace{\mathbf{V}^\top\mathbf{V}}_{\mathbf{I}}\mathbf{S}^\top\mathbf{U}^\top \quad \text{und} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{V}\mathbf{S}^\top\underbrace{\mathbf{U}^\top\mathbf{U}}_{\mathbf{I}}\mathbf{S}\mathbf{V}^\top \tag{A.3}$$

$$\Rightarrow \quad \mathbf{A}\mathbf{A}^\top\mathbf{U} = \mathbf{U}\mathbf{S}\mathbf{S}^\top \quad \text{und} \quad \mathbf{A}^\top\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{S}^\top\mathbf{S} \tag{A.4}$$

$$\text{mit} \quad \mathbf{I}: \text{ Einheitsmatrix}$$

Wenn man das **spezielle Eigenwertproblem** $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ betrachtet und berücksichtigt, dass $\mathbf{S}\mathbf{S}^\top$ bzw. $\mathbf{S}^\top\mathbf{S}$ Diagonalmatrizen sind, erkennt man, dass es sich in den Gleichungen (A.4) offenbar ebenfalls um ein Eigenwertproblem handelt. Die Spalten von $\mathbf{U}$ enthalten dabei die Eigenvektoren der symmetrischen Matrix $\mathbf{A}\mathbf{A}^\top$, entsprechendes gilt für $\mathbf{V}$. Die Singulärwerte sind also die positiven Wurzeln der Eigenwerte von $\mathbf{A}\mathbf{A}^\top$ oder $\mathbf{A}^\top\mathbf{A}$.

a. $\mathbf{A}$ hat vollen Rang



b. $\mathbf{A}$ hat keinen vollen Rang

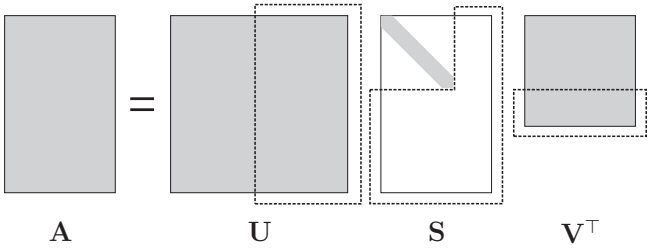


Abbildung A.1: Reguläre und reduzierte Form der SVD Zerlegung für Matrizen mit vollem und reduziertem Rang. Bei der reduzierten Form werden die gestrichelten Bereiche weggelassen.

A.1.3 Reduzierte Form der SVD

In Abb. A.1 ist der Aufbau der Matrizen veranschaulicht. Die Singulärwertmatrix $\mathbf{S}$ ist im gestrichelten Bereich Null. Daher können die eingerahmten Werte in $\mathbf{U}$ auch weggelassen werden, ohne Informationsverlust. Entsprechendes gilt bei einer Matrix $\mathbf{A}$ mit reduziertem Rang $r < l$ in Abb. b. Man erhält so die reduzierte Form der SVD. Aus der reduzierten Form folgt, dass man $\mathbf{A}$ auch als Summe von Matrizen des Rangs $r = 1$ schreiben kann:

$$\mathbf{A} = \sum_{j=1}^{r} s_j \mathbf{u}_j \mathbf{v}_j^\top \tag{A.5}$$

wobei u_j und v_j die Spalten von $\mathbf{U}$ bzw. $\mathbf{V}$ sind. In der Praxis fallen die Werte von s_j mit steigendem Index j schnell zu Null ab. Schneidet man die Summe in A.5 bei kleinen Singulärwerten ab, bietet dies eine sehr effektive Möglichkeit der Datenkompression [HMM01]

A.1.4 Anwendung der SVD auf Spektrogramme

Die SVD soll nun auf Spektrogramm-Daten angewendet werden. Die Spektrogramm Matrix $\mathbf{X}$ sei dabei so orientiert, dass sie die Dimension $T \times N$ habe, wobei T die Zeitdimension und N die Frequenzdimension sei. Die Singulärwertzerlegung liefert

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\mathbf{S}\mathbf{V}^\top \\ \text{mit } \mathbf{U} \in \mathbb{R}^{T\times T} &\quad \text{und } V \in \mathbb{R}^{N\times N} \end{aligned} \tag{A.6}$$

Die Eigenvektoren von $\mathbf{X}^\top\mathbf{X}$ in $\mathbf{V}$ sind zueinander orthogonal und haben den Betrag eins. Sie spannen damit einen orthogonalen $N \times N$-Raum auf, in welchen sich die Spektraldaten so transformieren lassen, dass sie durch eine Linearkombination der Eigenvektoren in $\mathbf{U}$ dargestellt werden können.

$$\mathbf{X} \cdot \mathbf{V} = \mathbf{U} \cdot \mathbf{S} \tag{A.7}$$

Damit ergibt sich folgender Aufbau der Matrizen:

$$\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^\top$$

$$\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ x_{31} & x_{32} & \dots & x_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T1} & x_{T2} & \dots & x_{TN} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1T} \\ u_{11} & u_{12} & u_{13} & \dots & u_{2T} \\ u_{11} & u_{12} & u_{13} & \dots & u_{3T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{T1} & u_{T2} & u_{T3} & \dots & u_{TT} \end{pmatrix} \cdot \left(\begin{array}{cccc} s_{11} & 0 & \dots & 0 \\ 0 & s_{22} & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_{NN} \\ \hline 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{array}\right)$$

$$\cdot \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1N} \\ v_{21} & v_{22} & \dots & v_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \dots & v_{NN} \end{pmatrix}^\top \tag{A.8}$$

Wird die Basisvektormatrix $\mathbf{V}$ nun auf die K wichtigsten (Basis-) Eigenvektoren entsprechend dem Betrag der zugehörigen Eigenwerte reduziert, so repräsentiert $\mathbf{V}_{\text{red}} \in \mathbb{R}^{N\times K}$ einen Unterraum von $\mathbf{V}$. Dieser enthält nun die „Richtungen“ der wichtigsten Spektralkomponenten in $\mathbf{X}$. Eine Multiplikation des Spektrogramms $\mathbf{X}$ mit der reduzierten Basisvektormatrix $\mathbf{V}_{\text{red}}$ entspricht geometrisch interpretiert einer Projektion der Datenwerte des Spektrogramms auf den von $\mathbf{V}_{\text{red}}$ aufgespannten Unterraum.

A.1.5 Deutung der SVD anhand einfach strukturierter Daten

Da die Projektion und Dimensionsreduktion nur schwer vorstellbar ist, soll der Vorgang noch einmal anhand einfach strukturierter Spektraldaten untersucht werden. Das betrachtete Signal besteht aus vier zeitlich aneinander gereihten Signalblöcken von Sinustönen:

Block	Sinus-Frequenz
1	800 Hz
2	600 Hz + 1000 Hz
3	500 Hz
4	800 Hz

Block 4 ist also eine Wiederholung von Block 1. Das Spektrogramm (Abb. A.2) eines derartigen Signals besteht dann aus vier dominanten horizontalen Linien an den entsprechenden Frequenzen.

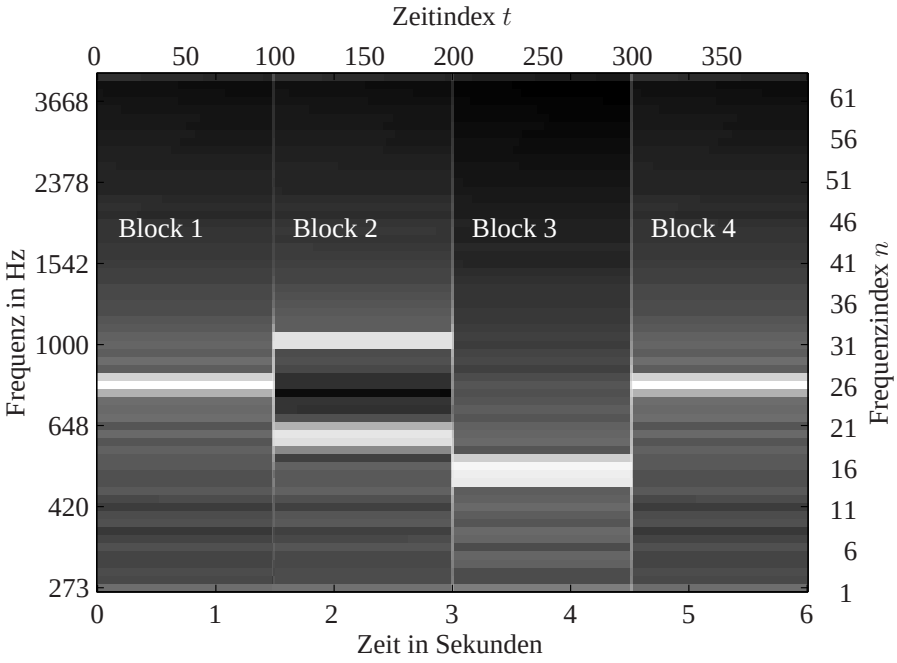


Abbildung A.2: Spektrogramm eines einfachen Signals aus Sinustönen

Betrachtet man das Spektrogramm als Aneinanderreihung von T Spektralvektoren der Länge N, erkennt man, dass einige Vektoren eine lineare Abhängigkeit zueinander haben. Mit der SVD wird eine Transformation derart vorgenommen, dass die resultierenden Vektoren linear unabhängig, und damit orthogonal sind. In Abb. A.2 gibt es in Frequenzrichtung nur 3 Vektoren die linear unabhängig sind. Diese werden ihrem Singulärwert-Betrag (in $\mathbf{S}$) nach sortiert in der

Matrix $\mathbf{V}$ gespeichert. Die Singulärvektoren $\mathbf{s}_n$ seien die Spaltenvektoren der Singulärwertmatrix $\mathbf{S}$:

$$\mathbf{s}_1 = (s_{11}, 0, 0, \ldots, 0)^\top \qquad \mathbf{s}_1 \in \mathbb{R}^{(T\times 1)} \tag{A.9}$$
$$\mathbf{s}_2 = (0, s_{22}, 0, \ldots, 0)^\top \qquad \text{usw.} \tag{A.10}$$

und die Basisvektoren $\mathbf{v}_n$ seien die Spaltenvektoren der Matrix $\mathbf{V}$:

$$\mathbf{v}_1 = (v_{11}, v_{12}, \ldots, v_{1N})^\top \qquad \mathbf{v}_1 \in \mathbb{R}^{(N\times 1)} \tag{A.11}$$
$$\mathbf{v}_2 = (v_{21}, v_{22}, \ldots, v_{2N})^\top \qquad \text{usw..} \tag{A.12}$$

Dann ergibt das Produkt aus $\mathbf{S}$ und $\mathbf{V}^T$:

$$\mathbf{S}\cdot\mathbf{V}^T = \begin{pmatrix} \mathbf{s}_1\mathbf{v}_1 & \mathbf{s}_1\mathbf{v}_2 & \ldots & \mathbf{s}_1\mathbf{v}_N \\ \mathbf{s}_2\mathbf{v}_1 & \mathbf{s}_2\mathbf{v}_2 & \ldots & \mathbf{s}_2\mathbf{v}_N \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_N 1\mathbf{v}_1 & \mathbf{s}_1\mathbf{v}_2 & \ldots & \mathbf{s}_N\mathbf{v}_N \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & \mathbf{1} & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ \mathbf{1} & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & \mathbf{1} & 0 & & 0 \\ 0 & \mathbf{1} & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & & 0 \end{pmatrix}^\top \begin{matrix} \vdots \\ 1200\ \text{Hz} \\ 1100\ \text{Hz} \\ 1000\ \text{Hz} \\ 900\ \text{Hz} \\ 800\ \text{Hz} \\ 700\ \text{Hz} \\ 600\ \text{Hz} \\ 500\ \text{Hz} \\ 400\ \text{Hz} \\ \vdots \end{matrix} \tag{A.13}$$

Die restlichen $N-3$ Spaltenvektoren von $\mathbf{V}$ sind also in diesem Fall bedeutungslos. Zwar ist auch hier die Orthogonalität der Vektoren erfüllt, aber die ihnen zugeordneten Singulärwerte in $\mathbf{S}$ sind so gering, dass man das Produkt $\mathbf{S}\cdot\mathbf{V}^T$ vernachlässigen kann. Bei diesem Signal kann man also praktisch ohne Informationsverlust eine Reduktion der Matrix $\mathbf{V}$ auf $\mathbf{V}_{\text{red}} \in \mathbb{R}^{N\times 3}$ vornehmen.
Mit der SVD enthält man noch eine weitere Matrix $\mathbf{U}$ der Dimension $T\times T$, in der die linear unabhängigen Zeilenvektoren in Zeitrichtung abgespeichert werden. Aus Abb. A.2 kann man entnehmen, dass auch in Zeitrichtung nur 3 Vektoren linear unabhängig sind. Das folgt auch daraus, dass der Spaltenrang einer $(N\times T)$-Matrix immer gleich dem Zeilenrang ist (Rang $r=3$) [BSMM93]. In Abb. A.3 sind die Matrizen ausschnittsweise dargestellt; alle nicht abgebildeten Werte sind Null.
Die Vektoren in $\mathbf{V}$ kann man auch als „spektrale Muster“ (z.B. von Harmonischen eines natürlichen Signals) auffassen. Das einfache Signal in Abb. A.2 enthält also $K=3$ unterschiedliche „spektrale Muster“.

Bezug zur Sprecherklassifikation Da jeder Sprecher durch die Individualität seines Vokaltraktes eine typische Formantstruktur hat, kann man annehmen, dass eine bestimmte Konstellation spektraler Muster einen Sprecher charakterisiert. Neben der Auswahl der $K<N$

wichtigsten Muster einer Sprecherklasse, also der ersten K Spaltenvektoren in $\mathbf{V}$, ist aber noch die Information darüber wichtig, *wann* ein solches Muster auftritt. Den zeitliche Verlauf kann man der Matrix $\mathbf{U} \cdot \mathbf{S}$ entnehmen (siehe Abb. A.3).

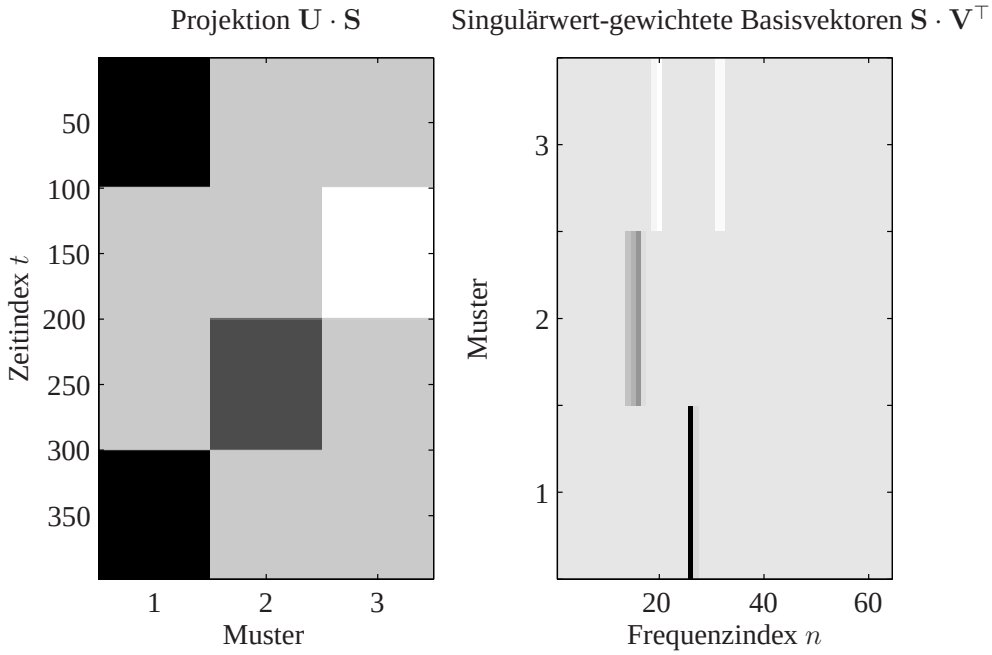


Abbildung A.3: Projektion und Basisvektoren für einfach strukturierte Spektraldaten

Muster 1 (die Spektrallinie bei 800 Hz) tritt am Anfang (0-100) und am Ende (300-350) auf, was an den schwarzen Balken in der ersten Spalte erkennbar ist. Muster 2 (500 Hz) tritt zwischen Zeitindex 200 und 300, Muster 3 (600 Hz +1000 Hz) zwischen 100 und 200 auf. Dies kann man direkt auch am Spektrogramm A.2 überprüfen.
Das Produkt $\mathbf{U} \cdot \mathbf{S}$ beinhaltet die in einen orthogonalen Raum transformierten Spektrogramm-Daten. Sind die Beträge der Singulärwerte gering oder werden „künstlich“ zur Dimensionsreduktion auf Null gesetzt, sind dies die Projektionsdaten von $\mathbf{X}$ auf $\mathbf{V}_{\mathrm{red}}$. Es ist daher für die Klassifikation nur nötig, $\mathbf{V}$ (bzw. $\mathbf{V}_{\mathrm{red}}$) abzuspeichern, also die Information über die (wichtigsten) spektralen Muster.

Anhang B

Definitionen

B.1 Mathematische Definitionen

Ereignisse in einem vollständigen Ereignissystem

Wenn **A** eine Ereignismenge ist und die Ereignisse $B_i \in \mathbf{A}$ mit $P(B_i) > 0 (i = 1, 2, \ldots n)$ ein vollständiges Ereignissystem bilden, dann gelten für jedes Ereignis $A \in \mathbf{A}$ die folgenden Sätze [BSMM93]:

Satz der vollständigen Wahrscheinlichkeit Die Wahrscheinlichkeit für ein Ereignis A ist gleich dem Produkt aus den Wahrscheinlichkeiten der Ereignisse B_i und der Summe aller bedingten Wahrscheinlichkeiten $P(A|B_i)$:

$$P(A) = \sum_{i=1}^{n} P(A|B_i)P(B_i). \tag{B.1}$$

Satz von Bayes Die Wahrscheinlichkeit für ein Ereignis B_k unter der Bedingung, dass das Ereignis A eingetreten ist, ist gleich dem Produkt aus der bedingten Wahrscheinlichkeit $P(A|B_k)$ und der Wahrscheinlichkeit von B_k geteilt durch die vollständige Wahrscheinlichkeit $P(A)$ aus (B.1):

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{P(A)} \tag{B.2}$$

Erwartungswert einer diskreten Zufallssvariable Im diskreten Fall ergibt sich als Erwartungswert der Zufallsgröße X das *gewogene Mittel*:

$$\mathrm{E}\{X\} = p_1 x_1 + p_2 x_2 + \ldots p_n x_n \tag{B.3}$$

der Werte $x_1, \ldots, x_n$ mit den Wahrscheinlichkeiten $p_k(k = 1 \ldots n)$, Gewichte genannt. Bei der Gleichverteilung ist $p_1 = p_2 = \ldots = p_n = 1/n$, und $\mathrm{E}\{X\}$ wird zum arithmetischen Mittel der Werte x_k.

Fenstergewichtung Um von einem Zeitsignal mit Hilfe der Kurzzeit-Fouriertransformation eine Zeit-Frequenz-Darstellung (Spektrogrammschätzung) zu erhalten, muss dieses Signal in zeitliche begrenzte Abschnitte, „Zeit-Fenster“, geteilt werden. Da eine Rechteck-Fensterung im Spektralbereich zum Leck-Effekt führt [KK98], ist es günstiger, andere zeitbegrenzende Fensterfunktionen zu verwenden. Eine der gebräuchlichsten Fensterfunktionen ist das **Hamming-Fenster:**

$$f^{Hm}(k) = \begin{cases} 0.54 - 0.46 \cdot \cos(2\pi k/(N-1)), & 0 \leq k \leq N-1 \\ 0, & \text{sonst} \end{cases} \tag{B.4}$$

Die Hamming-Fensterfunktion minimiert das Hauptmaximum im Sperrbereich.

Anhang C

Akronyme und Abkürzungen

Allgemeine Notation:

- Vektoren werden als fett gedruckte Kleinbuchstaben dargestellt. Wo nicht anders definiert, handelt es sich um Spaltenvektoren.
  Beispiel: $\mathbf{x} = [x_1, x_2, \ldots, x_T]^\top$
  $^\top$ bezeichnet die Transposition eines reellen Vektors.
- Matrizen werden als fett gedruckte Großbuchstaben dargestellt.
  Beispiel: $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}]$

ASB	Audio Spectrum Basis (MPEG-7), Basisvektormatrix $\mathbf{V}$ bzw. reduzierte Form $\mathbf{V}_{\mathrm{red}}$
ASE	Audio Spectrum Envelope (MPEG-7), Audio-Spektrogramm mit spezieller, logarithmierter Frequenzachse
ASP	Audio Spectrum Projection (MPEG-7), Projektion des normierten Spektrums auf die Basisvektoren
dB	Dezibel
DCT	Diskrete Cosinus Transformation
DFT	Diskrete Fourier Transformation, (sowohl zeit- als auch frequenzdiskret)
EM	Expectation-Maximization
FFT	Fast Fourier Transform, schnelle Fouriertransformation
FIR	Finite Input Response (Filter), nichtrekursives und daher zeitbegrenztes Filter
GMM	Gaussian Mixture Model, Gaußsches Mischverteilungsmodell
GMM/D	GMM mit diagonalen Kovarianzmatrizen bzw. Varianzen
GMM/F	GMM mit vollen Kovarianzmatrizen

HMM	Hidden Markov Model
IDCT	Inverse DCT
IDFT	Inverse DFT
MFCC	Mel Frequency Cepstral Coefficients, auf eine Mel-Frequenz-Skala basierende Cepstral-Koeffizienten
MPEG	Motion Picture Expert Group
MPEG-7	MPEG-7 ist ein neuer internationaler Standard für die Beschreibung des Inhalts von Medien, der gut geeignet ist für Anwendungen wie Musik-Indizierung, Ähnlichkeitsprüfung, und wissensbasierter Audio-Verarbeitung. Im Standard sind unter anderem Verfahren zu Merkmal-extraktion, Ähnlichkeitsbeschreibung und Klassifikation definiert.
SVD	Singular Value Decomposition, Singulärwertzerlegung (s. Anhang A.1)
STFT	Short Time Fourier Transformation, Kurzzeit-Fouriertransformation
VQ	Vektorquantisierer

Formelzeichen

$s(k)$	diskretes Zeitsignal
$\mathbf{X}$	Observationsmatrix
$\mathbf{P}$	Audio Spectrum Projection
$\mathbf{V}, \mathbf{V}_{\mathrm{red}}$	Audio Spectrum Basis
$\mathbf{\Phi_{lin}}$	Spektrogramm aus linearen FFT-Koeffizienten
$\mathbf{\Phi_{log}}$	ASE-Spektrogramm mit logarithmischer Frequenzachse
$\boldsymbol{\psi}_{\mathbf{env}}$	Einhüllende des Spektrogramms
$\widetilde{\mathbf{\Psi}}$	Normiertes Spektrogramm
n_{FFT}	Länge der FFT
$\mathbf{LT}$	Transformationsmatrix
f_s	Abtastfrequenz des Audio-Signals
n	Frequenzindex
N	Anzahl der Frequenzkoeffizienten, Frequenzdimension
t	Zeitindex
T	Anzahl der Analyse-Zeitrahmen, Zeitdimension
M	Modell-Ordnung
p_i	Mischungsgewichte (mixture weights)
μ_i	Mittelwerte, bzw. Mittelwertsvektoren
σ_i^2	Varianzen, bzw. Varianzvektoren
$\mathbf{C_{xx_i}}$	Kovarianzmatrizen
λ	Modellparameter

Literaturverzeichnis

[BE67] L. E. Baum und J. A. Eagon. *An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model of ecology. Bull. Amer. Math. Soc.*, 73, Seite 360–363, 1967. 3.2

[BS68] L.E. Baum und G.R. Sell. *Growth functions for transformations on manifolds*. *Pac. J. Math.*, 27 (Nummer 2), Seite 211–227, 1968. 3.2

[BSMM93] I. Bronstein, K. Semendjajew, G. Musiol, und H. Muehlig. *Taschenbuch der Mathematik*. Harri Deutsch Verlag, Thun, 1993. A.1.5, B.1

[Car98] J.-F. Cardoso. Multidimensional independent component analysis. In *Proc. ICASSP '98. Seattle*, 1998. 3.1.6

[Cas01] M. Casey. *General Sound Classification and Similarity of MPEG-7 Audio*. *Organized Sound*, August 2001. 3.1.6

[DLR77] A. P. Dempster, N. M. Laird, und D. B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. B*, 39, Seite 1–38, 1977. 3.1.1

[Fen99] M. Feng. *Blinde Signalverarbeitung mit Hilfe von Sensorgruppen für mobile Kommunikationssysteme*. Shaker Verlag, Aachen, Bremen, 1999. 3.1.6

[HKO01] A. Hyvärinen, J. Karhunen, und E. Oja. *Independent Component Analysis*. John Wiley & Sons, Inc., N.Y., 2001. 3.1.6

[HMM01] B. Herbst, N. Muller, und L. Magaia. *The singular value decomposition and facial recognition*. *SIAM Review*, 2001. A.1.3

[Hän97] E. Hänsler. *Statistische Signale*. Springer Verlag, 1997. 3.1.1, 3.3

[ISO02] ISO/IEC. *MPEG-7 Multimedia Description Scheme - Part 4 Audio*. ISO/IEC 15398-4:2002, 2002. 2.1, 2.2.2

[KK98] K.D. Kammeyer und K. Kroschel. *Digitale Signalverarbeitung*. B.G. Teubner, Stuttgart, 1998. B.1

[LBG80] Y. Linde, A. Buzo, und R. M. Gray. *An algorithm for vector quantizer design*. *IEEE Trans. on Communications*, 28 (Nummer 1), Seite 84–95, 1980. 3.3, 3.3.1

[MP00] G. McLachlan und D. Peel. *Finite Mixture Models*. Wiley series in probability and statistics. John Wiley & Sons, Inc., 2000. 3.1.1, 3.1.3

[Rab89] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. *Proceedings of the IEEE*, 77 (Nummer 2), Seite 257–285, 1989. 3.2, 3.2, 3.2

[RJ93] R. A. Reyment und K. G. Joreskog. *Applied Factor Analysis in the Natural Sciences*. Cambridge University Press, Cambridge, UK, 1993. 2.2.1

[RR95] D. A. Reynolds und R. C. Rose. *Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*. *IEEE Trans. on Speech and Audio Processing*, 3, Seite 72–83, January 1995. 3.1.5, 3.1.6, 5.2.4

[Vac90] R. J. Vaccaro. *SVD and Signal Processing, II*, Amsterdam, 1990. Elsevier Science Publishers. 2.2.1

[VHH98] P. Vary, U. Heute, und W. Hess. *Digitale Sprachsignalverarbeitung*. Informationstechnik. Teubner, Stuttgart, first Auflage, 1998. 2.3

[Vit67] A. Viterbi. *Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm*. *IEEE Trans. Information Theory*, IT-13, Seite 260–269, 1967. 3.2.1