

Bachelorarbeit

**Strukturelle Komplexitätsmerkmale zur
Musikgenre-Klassifikation mit
Fuzzy- k -nearest-Neighbors**

Philipp Ginsel

1. Oktober 2018

überarbeitete Version vom 22. Januar 2019

Betreuer:

Prof. Dr. Günter Rudolph

Dr. Igor Vatulkin

Fakultät für Informatik

Algorithm Engineering (LS 11)

Technische Universität Dortmund

<http://ls11-www.cs.tu-dortmund.de>

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Hintergrund	1
1.2	Ziele der Arbeit	2
1.3	Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Merkmalsextraktion	5
2.2	Merkmalsverarbeitung	9
2.2.1	Ersetzung fehlender Werte	11
2.2.2	Onset-Pruning	11
2.2.3	Normalisierung	12
2.2.4	Strukturelle Komplexität	13
2.2.5	Vorbereitung von Klassifikationsinstanzen	14
2.2.6	Merkmalsauswahl	15
2.3	Klassifikation	17
2.3.1	Überwachte Klassifikation	17
2.3.2	Der Fuzzy- k -nearest-Neighbors-Algorithmus	18
2.4	Bewertung	21
2.4.1	Resampling	21
2.4.2	Bewertungsmaße	22
3	Experimente	25
3.1	Versuchsaufbau	25
3.1.1	Testdaten	26
3.1.2	Merkmalsauswahl mit Relief	27
3.2	Ergebnisse	28
3.2.1	Strukturelle Komplexität	28
3.2.2	FKNN	41
3.2.3	Relief-Merkmalsauswahl	46
4	Zusammenfassung und Ausblick	53

A Anhang	57
A.1 Merkmalsgruppen	57
A.2 Balancierter relativer Fehler	59
Abbildungsverzeichnis	76
Algorithmenverzeichnis	77
Literaturverzeichnis	81
Eidesstattliche Versicherung	81

Kapitel 1

Einleitung

Durch das große Angebot an Streaming- und Download-Diensten für Musik wird *Music Information Retrieval* (MIR), also das automatische Analysieren von Musikdaten, immer wichtiger. So kann es zum Beispiel wichtig sein, einem Nutzer neue Musik vorzuschlagen oder Musik automatisch in Kategorien oder Playlisten einzuteilen. Nicht nur die Nachfrage nach derartigen Methoden steigt. Mit der großen Menge an Musikstücken, die online verfügbar sind und über die bereits viele Informationen, wie Genre, Künstler, Vorkommen in Playlisten oder Hörverhalten der Nutzer bekannt und digital verfügbar sind, wird es auch immer einfacher, neue Methoden für verschiedene Aufgaben des MIR zu testen.

1.1 Motivation und Hintergrund

Eine häufige Problemstellung von MIR ist es, Musikstücke in Genres zu klassifizieren. Genres sind Kategorien, in die man verschiedene Arten von Musik einteilt, wie z.B. Klassik, Pop, Rock oder Jazz. Der Vorteil der Genre-Klassifikation ist es, dass im Gegensatz zu persönlichen Musikvorlieben, bereits große Mengen von in Genres unterteilter Musik zur Verfügung stehen, so wie zum Beispiel die Datenbanken der AllMusicGuide-Webseite.¹ Eine Schwierigkeit der Genreklassifikation ist es, dass es sehr schwierig ist verschiedene Genres von Musik zu definieren. So kann es sein, dass ein Musikstück nur schwierig einem Genre zugeordnet werden kann oder mehreren Genres angehört.

Aus diesen Gründen ist für dieses Problem möglicherweise *Fuzzy-Klassifikation* sinnvoll. Also Klassifikationsmethoden zu verwenden, die auf Fuzzy-Logik oder Fuzzy-Mengen beruhen [24]. Diese Methoden entscheiden nicht fest, ob ein Stück zu einem Genre gehört, sondern ordnen ihm einen Zugehörigkeitsgrad zu diesem Genre zu, der beschreibt, wie stark dieses Stück diesem Genre angehört bzw. wie sicher man sich bei dieser Zuordnung sein kann. Ein Algorithmus, der für diese Art von Fuzzy-Klassifikation verwendet werden kann, ist der Fuzzy- k -nearest-Neighbors-Algorithmus (FKNN), der 1985 von Keller, Gray und

¹<http://www.allmusic.com/> (zuletzt geprüft am 23.09.2018 um 21:18 Uhr)

Givens, als eine fuzzy Variante des klassischen k -nearest-Neighbor-Algorithmus entwickelt wurde [8]. Dort zeigte sich, dass die Zugehörigkeitsgrade, die der Algorithmus berechnet, gut beschreiben, mit welcher Sicherheit eine Klassifikation korrekt ist. Auf diese Weise lassen sich seine Klassifikationsergebnisse besser deuten als die Ergebnisse eines Klassifikators, der nicht mit solchen Zugehörigkeitsgraden arbeitet. FKNN wurde in [1] und [23] bereits dazu verwendet, Emotionen in Musik zu erkennen und führte dort zu guten Ergebnissen. Eine andere Methode für Fuzzy-Klassifikation sind *regelbasierte Fuzzy-Systeme*, die *Fuzzy-Regeln*, eine unscharfe Art von logischen Regeln, zur Klassifikation verwenden. Diese Art von Systemen wurden z.B. in [5] und [18] verwendet um Musikgenres zu klassifizieren. Da nach unserem Wissen FKNN noch nicht zur Klassifikation von Musikgenres verwendet wurde, möchten wir ihn in dieser Arbeit gerne für diese Aufgabe untersuchen.

Ein wichtiger Teil der Klassifikation von Musik ist, auszuwählen, welche Merkmale der Musik man verwendet und wie man diese verarbeitet. Eine interessante Methode musikalische Merkmale zu verarbeiten ist die *strukturelle Komplexität*, die 2011 von Mauch und Levy in [12] vorgestellt wurde. Es beschreibt wie stark sich Merkmale von Musik auf verschiedenen Zeitskalen verändern. Eine Untersuchung von Mauch und Levy zeigt, dass dieses Maß mit der von Menschen empfundenen Komplexität von Musik zusammenhängt [12]. Die strukturelle Komplexität ist eine Methode, die in der Lage ist, mehrere musikalische Merkmale über verschiedene Zeiträume zusammenzufassen. Dabei ist es sinnvoll Gruppen musikalischer Merkmale zu bilden, die ähnliche Eigenschaften der Musik beschreiben, und auf ihnen die strukturelle Komplexität zu berechnen. So kann man z.B. alle Merkmale, die Eigenschaften über den Rhythmus beschreiben zu einer Gruppe zusammenfassen. Die strukturelle Komplexität dieser Gruppe, beschreibt dann die rhythmische Komplexität von Musik. Auf diese Weise kann man im besten Fall verarbeitete Merkmale erhalten, die leicht verständliche und gut interpretierbare Eigenschaften von Musik beschreiben. Dieser Ansatz wurde bereits in [16] in Kombination mit anderen Methoden verwendet um Musikgenres zu klassifizieren. In dieser Arbeit wollen wir uns noch etwas direkter auf diese Methode konzentrieren und herausfinden, welche Einstellungen sich gut zur Klassifikation von Musikgenres eignen.

1.2 Ziele der Arbeit

Wir möchten in dieser Arbeit die Klassifikation von verschiedenen Genres auf einer großen Mengen von Musikstücken mit strukturellen Komplexitätsmerkmalen (s. Abschnitt 2.2.4) und dem FKNN-Algorithmus (s. Abschnitt 2.3.2) testen.

Dabei möchten wir herausfinden, welche Einstellungen der strukturellen Komplexität sich besonders gut zur Klassifikation von Genres allgemein und für konkrete Genres eignen. Die strukturelle Komplexität berechnet die Änderung von verschiedenen Gruppen von verschiedenen musikalischen Merkmalen auf verschiedenen Zeitskalen, um die Kom-

plexität verschiedener musikalischer Aspekte eines Stücks zu beschreiben. Dafür werden Gruppen von Merkmalen über Fenster unterschiedlicher Größe zusammengefasst und Abstände dieser Zusammenfassungen berechnet. Wir möchten in dieser Arbeit testen, welche Merkmalsgruppen und welche Fenstergrößen sich besonders gut zur Klassifikation der unterschiedlichen Genres eignen.

Für den FKNN-Algorithmus interessiert uns, welche Werte für den Parameter k , der bestimmt, wie viele der nächsten Nachbarn einer zu klassifizierenden Instanz zu ihrer Klassifikation genutzt werden, zu den besten Ergebnissen führen. Außerdem interessiert uns, ob die Zugehörigkeitsgrade, die der Algorithmus berechnet, wirklich die Sicherheit der Klassifikation widerspiegeln, so wie es Keller, Gray und Givens in [8] beschreiben.

Außerdem möchten wir testen, ob eine einfache Merkmalsauswahlmethode wie Relief (s. Abschnitt 2.2.6), die Klassifikation mit strukturellen Komplexitätsmerkmalen und FKNN verbessern kann, und welche Merkmale sie dazu auswählt.

1.3 Aufbau der Arbeit

Diese Arbeit besteht aus insgesamt vier Kapiteln. Nachdem wir in diesem ersten Kapitel eine Einleitung in den Inhalt dieser Arbeit gegeben haben, werden im folgenden zweiten Kapitel die Grundlagen beschrieben, die nötig sind um die Untersuchungen dieser Arbeit nachzuvollziehen. Es wird beschrieben, wie Musikgenreklassifikation mit den in dieser Arbeit verwendeten Methoden möglich ist und wie man die Leistung dieser Methoden bewerten kann. Im daran anschließenden dritten Kapitel beschreiben wir dann, welche Experimente im Rahmen dieser Arbeit durchgeführt werden, um unsere Ziele zu erreichen, und zu welchen Ergebnissen diese Experimente geführt haben. Im vierten und letzten Kapitel werden dann die wichtigsten Ergebnisse und Erkenntnisse dieser Arbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Forschung gegeben.

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen vorgestellt, die für die Experimente nötig sind, die im Rahmen dieser Arbeit durchgeführt werden. Wir beschreiben, wie wir Merkmale der Musik extrahieren, verarbeiten und nutzen, um Musikgenres zu klassifizieren.

2.1 Merkmalsextraktion

Um Musik zu analysieren und zu klassifizieren, werden Merkmale benötigt, die sie beschreiben. Wir verwenden in dieser Arbeit dazu Audiomerkmale, das heißt Merkmale, die direkt aus dem Audiosignal extrahiert werden, im Gegensatz zu Merkmalen, die aus anderen Quellen stammen, wie z.B. MIDI-Dateien [13, S. 171], oder Playlisten [17].

Um zu beschreiben, wie das funktioniert, müssen wir zunächst erklären, wie ein Audiosignal digital dargestellt wird. Dazu wird in regelmäßigen Abständen die Amplitude des Signals gemessen. Um diese Messwerte zu speichern, müssen sie noch diskretisiert werden, das heißt auf einen darstellbaren Wertebereich abgebildet werden. Will man die Abtastwerte z.B. als 16Bit-Werte speichern, muss man also die theoretisch unendlich verschiedenen möglichen Amplituden auf 2^{16} diskrete Werte abbilden [22, S. 186–188]. Die Anzahl der Abtastwerte, die pro Sekunde gemessen werden, nennt man *Abtastrate*. Die Audiodaten in einer Datei sind im Grunde nur eine lange Liste der Abtastwerte. Wenn man die Abtastrate und die Darstellung der Messwerte kennt, kann man daraus dann das ursprüngliche Signal approximieren [22, S. 186]. Dabei ist es möglich Frequenzen wiederherzustellen, die maximal die Hälfte der Abtastrate betragen [7]. Auf einer Audio-CD, die eine Abtastrate von 44100Hz hat, können also Frequenzen von bis zu $\frac{44100\text{Hz}}{2} = 22,05\text{kHz}$ gelesen werden. Da Menschen Töne zwischen ca. 20Hz und 20kHz hören können, wird damit das gesamte menschliche Gehör abgedeckt [14, S. 80].

Aus diesen Audiodaten werden dann die Merkmale berechnet, die die Musik beschreiben und zur Klassifikation genutzt werden sollen. Die Merkmale werden dabei auf Fenstern berechnet, die aus mehreren Abtastwerten bestehen. Die Größen dieser Fenster sind von

Merkmal zu Merkmal unterschiedlich. Zum Beispiel wird ein Merkmal, das bei einer Abtastrate von 44100Hz und einer Fenstergröße von 512 berechnet wird, für Zeitfenster der Länge $\frac{512}{44100\text{Hz}} \approx 12\text{ms}$ berechnet. Eine häufig angewendete Methode, um Merkmale zu berechnen, ist die Fouriertransformation [22, S. 31–34]. Sie berechnet aus den Audiodaten, welche den Verlauf des Signals über die Zeit beschreiben, die Stärke der verschiedenen Frequenzen, die im Signal vorkommen und deren Phasenverschiebung. Viele Merkmale werden dann entweder auf dem Verlauf des Signals über die Zeit (Zeitbereich), oder auf den Stärken der verschiedenen Frequenzen (Frequenzbereich) berechnet. So kann man z.B. auf dem Zeitbereich – also ohne die Fouriertransformation – das Merkmal *Zero-crossing rate* berechnen, welches misst, wie oft das Signal durch die x-Achse läuft. An ihm lassen sich Informationen über die Klangfarbe ablesen. Auf dem Frequenzbereich kann man hingegen Merkmale wie *Chroma* berechnen, welches angibt, wie stark die zwölf Halbtöne der westlichen Musik zu einem bestimmten Zeitpunkt zu hören sind.

Da viele Merkmale, wie z.B. Chroma, mehrdimensional sind, ergibt sich nach der Extraktion für jedes Merkmal eine Matrix, bei der die Reihen für die verschiedenen Dimensionen des Merkmals und die Spalten für die Zeitfenster stehen, für die sie berechnet werden. Da verschiedene Merkmale unterschiedliche Fenstergrößen haben, sind die Merkmalsmatrizen der Merkmale auch verschieden groß. Zur Extraktion verwenden wir das in [21] vorgestellte AMUSE-Framework.

Wir teilen die Merkmale in dieser Arbeit in die folgenden sieben Gruppen ein, die wir aus [16] übernommen haben und für das Verfahren der strukturellen Komplexität verwenden (s. Abschnitt 2.2.4):

- **Akkorde**

Die Merkmalsgruppe *Akkorde* besteht aus drei Merkmalen, die in [16] vorgestellt werden und mithilfe des Chordino Vamp Plugins [11] berechnet werden, welches zu jedem Zeitpunkt den aktuellen Akkord berechnet. Die drei Merkmale werden für Zeitfenster der Größe 10s berechnet. Sie beschreiben, wie viele verschiedene Akkorde in den 10s gespielt wurden, wie häufig die Akkorde in der Zeit gewechselt haben und wie stark die häufigsten Akkorde des Stücks in diesem 10s Abschnitt vertreten sind.

- **Chroma**

Die Merkmalsgruppe *Chroma* besteht aus den Merkmalen *Chroma* und *Bass-Chroma*. Chroma ist ein zwölf-dimensionales Merkmal, welches die Stärke der zwölf Halbtöne angibt, die in der westlichen Musik vorkommen. Das Merkmal Chroma deckt dabei alle Frequenzen der Musik ab, während sich Bass-Chroma auf einen bestimmten Bereich niedrigerer Frequenzen beschränkt [11].

- **Chroma-verwandt**

Die Merkmalsgruppe *Chroma-verwandt* besteht aus Merkmalen, die den Chroma-

merkmalen ähnlich sind oder aus diesen berechnet werden. So enthält sie Informationen über Amplitude, Position und Breite der fünf höchsten Peaks im Frequenzbereich und den Betrag und die Position des größten Chromawertes [15]. Außerdem enthält die Gruppe Merkmale, die die Grundfrequenz [15], die Inharmonizität [10] und die Änderung der Harmonien beschreibt [10].

- **Harmonie**

Die Merkmalsgruppe *Harmonie* setzt sich aus einer großen Menge von Merkmalen zusammen, die verschiedene Eigenschaften der Harmonie beschreiben. Sie enthält die zuvor beschriebenen Akkordmerkmale und auch die Merkmale über das Maximum des Chromavektors und dessen Position aus der Merkmalsgruppe Chroma-verwandt. Außerdem enthält die Gruppe noch einige Informationen über Intervalle, Tonarten und Stimmungen.

- **Instrumente**

Die Merkmalsgruppe *Instrumente* besteht aus acht Merkmalen, die für Zeitfenster von jeweils 10s Länge bestimmen, ob eine bestimmte Instrumenten Gruppe vertreten ist. Dazu wurden die Klassifikatoren *Random Forest* und *Support Vector Machine* auf Akkorden, die aus mehreren Instrumenten bestehen, trainiert, die Instrumentengruppen Gitarre, Piano, Blasinstrumente und Streichinstrumente zu erkennen. Für jede der vier Instrumentengruppen wird dann das beste Modell des Random Forest und der SupportVector Machine zur Berechnung der Merkmale benutzt. So ergeben sich acht Merkmale, die das Vorkommen dieser Instrumentengruppen beschreiben.

- **Tempo und Rhythmus**

Die Merkmalsgruppe *Tempo und Rhythmus* enthält Merkmale, die die Anzahl von Beats, Tatum und Onsets pro Minute messen [15]. Beats sind die am deutlichsten betonte musikalische Zeiteinheit. Sie entsprechen der Zeiteinheit, die z.B. ein Dirigent anzeigen würde [2]. Tatum sind dagegen die kleinste musikalische Einheit [2]. Als Onsets bezeichnet man Zeitpunkte, zu denen ein Ton beginnt. Ein weiteres Merkmal dieser Gruppe ist das Tempo, welches mithilfe der Onsetzeiten berechnet wird [10]. Die restlichen Merkmale beschreiben rhythmische Fluktuationen [10] [15] und die Deutlichkeit des Rhythmus [10]. In [16] wurde auch die Länge des Musikstücks als Merkmal in dieser Gruppe verwendet, da es für manche Genres eine wichtige Aussagekraft haben kann. Wir verwenden es in dieser Arbeit nicht, da wir zur Klassifikation nur die strukturelle Komplexität der Merkmale verwenden (s. Abschnitt 2.2.4). Dabei wird die Änderung der Werte einer Merkmalsgruppe über die Zeit berechnet. Da sich die Länge eines Musikstücks nicht ändert, hat sie auch keinen wirklichen Einfluss auf die strukturelle Komplexität.

- **Klangfarbe**

Die Merkmalsgruppe *Klangfarbe* setzt sich aus einer großen Menge von 29 Merkmalen zusammen, die viele verschiedene Eigenschaften über Amplituden, Verteilung von Frequenzen und Phasenverschiebungen beschreiben. Diese Merkmale beschreiben alle verschiedene Aspekte der Klangfarbe.

Die genaue Zusammensetzung dieser Gruppen findet man im Anhang A.1.

Zusätzlich zu diesen sieben Gruppen benutzen wir das Merkmal *Akkordvektor*, das wir in dieser Arbeit einführen. Wir verwenden es als achte Merkmalsgruppe, die aus nur einem Merkmal besteht. Die Merkmalsverarbeitungsmethode strukturelle Komplexität, die wir in dieser Arbeit verwenden, beschreibt die Änderung von Merkmalsgruppen auf unterschiedlichen Fenstergrößen. Auf diese Weise beschreibt es die Veränderung einer Merkmalsgruppe über die Zeit (s. Abschnitt 2.2.4). Da die Merkmale der Merkmalsgruppe Akkorde bereits die Änderung der Akkorde beschreiben, sorgt das dafür, dass diese Merkmalsgruppe nach der Verarbeitung in gewisser Weise die Änderung der Änderung beschreibt. Außerdem erlaubt die Fenstergröße dieser Akkordmerkmale es nur Änderungen zu erkennen, die über einen Zeitraum von über 10s stattfinden. Deshalb führen wir das Merkmal *Akkordvektor* ein, das die vom Chordino Vamp Plugin [11] berechneten Akkorde auf eine Weise darstellt, die für die strukturelle Komplexität nutzbar ist. Das Chordino Vamp Plugin berechnet für ein Musikstück für jeden Zeitpunkt den aktuell gespielten Akkord und stellt ihn als Akkordsymbol dar. Da die strukturelle Komplexität Mittelwert und Abstände von Vektoren berechnet, müssen wir diese Akkordsymbole so in numerische Vektoren umwandeln, dass der Mittelwert mehrerer Akkordvektoren und der Abstand dieser Mittelwerte eine sinnvolle Bedeutung haben. Wir wandeln jedes Akkordsymbol in einen 20-dimensionalen binären Vektor um. Die ersten zwölf Stellen des Vektors beschreiben dabei den Grundton des Akkords. Jede Stelle steht für einen Ton und genau eine dieser Stellen hat den Wert 1. Die nächsten vier Stellen stehen für die Art des Akkords. Das Chordino Vamp Plugin unterscheidet zwischen Dur-, Moll-, verminderten und übermäßigen Akkorden. Jede der vier Stellen steht für eine dieser Arten und es hat immer genau einer dieser Stellen den Wert 1. Die letzten vier Stellen beschreiben mögliche Erweiterungen des Akkords. Das Chordino Vamp Plugin kann dabei kleine Septimen, große Septimen, verminderte Septimen und Sexten erkennen und gibt pro Akkord immer nur eine Erweiterung an. Jede der vier Stellen steht für eine der Erweiterungen. Wenn der Akkord eine Erweiterung hat, hat genau eine dieser Stellen den Wert 1, ansonsten haben sie alle den Wert 0. Könnte das Plugin keinen Akkord erkennen, bzw. hat erkannt, dass zu diesem Zeitpunkt kein Akkord gespielt wird, haben alle 20 Stellen den Wert 0. Auf diese Weise können wir alle Informationen, die uns das Plugin liefert in unserem Vektor darstellen, nur den Basston nicht, der manchmal von dem Plugin mitangegeben wird. Darauf verzichten wir, da wir den Vektor sonst deutlich größer hätten machen müssen ohne viele Informationen hinzuzugewinnen. Die zwei

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	Dur	Moll	Verm.	Übern.	kl. Sept	gr. Sept.	verm. Sept.	Sexte
C	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Cm	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Cdim	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Caug	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
C ⁷	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
Cm ⁷	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
Cmaj ⁷	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
Cdim ⁷	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
C ⁶	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
G ⁷	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0
G ⁷ /B	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0
D#	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Eb	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabelle 2.1: Darstellung der vom Chordino Vamp Plugin extrahierten Akkordsymbole als binäre Vektoren

Akkorde G^7 und G^7/B führen bei unserem Merkmal also zu gleichen Werten. Außerdem unterscheiden wir nicht zwischen enharmonischen Verwechslungen, wie $D\#$ und Eb , da sie gleich klingen [25, S. 42]. In der Tabelle 2.1 demonstrieren wir die Darstellung einiger Beispielakkorde. Wir verwenden die international verbreitete englische Akkordschreibweise, die auch vom Plugin genutzt wird. N steht dabei dafür, dass kein Akkord erkannt wurde.

Auf diese Weise haben wir ein Merkmal entwickelt, bzw. eine Darstellung des Chordino-Vamp-Plugin-Merkmals entwickelt, die sich dazu eignet mithilfe von struktureller Komplexität die Komplexität der Akkorde eines Stücks zu bestimmen. Der Mittelwert mehrerer Akkorde hat dabei eine sinnvolle Bedeutung. Jede Stelle beschreibt den Anteil der Akkorde, die einen bestimmten Grundton, Art oder Erweiterung haben. Der Abstand zwischen Durchschnitten zu unterschiedlichen Zeitpunkten lässt sich dann als die Änderung der Zusammensetzung der Akkorde des Stücks deuten.

2.2 Merkmalsverarbeitung

Nachdem die Merkmale in Form von Merkmalsmatrizen extrahiert worden sind, müssen sie weiter zu Instanzen verarbeitet werden, die von einem Klassifikationsalgorithmus verwendet werden können. Dies ist die Aufgabe der *Merkmalsverarbeitung*.

Davor müssen jedoch die Merkmalsmatrizen der einzelnen Merkmale zu einer großen Merkmalsmatrix zusammengesetzt werden. Die Schwierigkeit dabei ist, dass die Extraktionsfenster der verschiedenen Merkmale sehr unterschiedliche Längen haben. So wird das Chroma z.B. für Zeitfenster mit einer Größe von ca. 92,9ms extrahiert, während die Instrumentenmerkmale für Zeitfenster der Größe 10s extrahiert werden. Das bedeutet auch, dass bei einem Musikstück er Länge 45s, die Instrumentenmerkmale in den letzten 5s gar nicht extrahiert werden können. Es bedarf also einer Methode, die Merkmalsmatrizen mit unter-

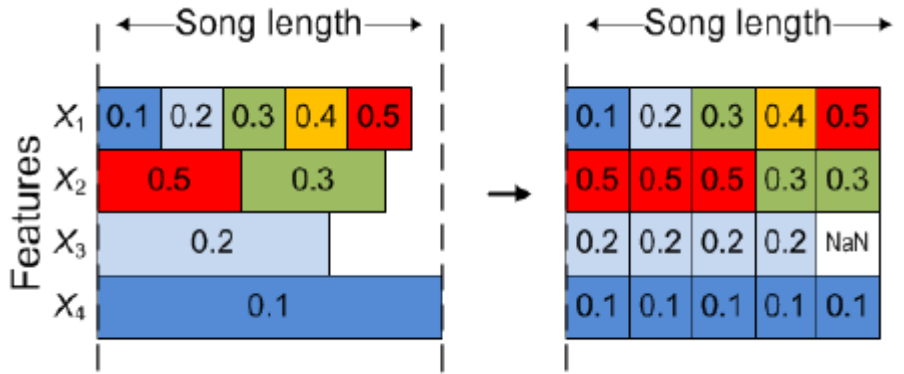


Abbildung 2.1: Erzeugung der harmonisierten Merkmalsmatrix X^H [16, S. 34]

schiedlichen Fenstergrößen so zusammensetzt, dass die Merkmalswerte, die zum gleichen Zeitpunkt eines Stückes gehören, danach in der gleichen Spalte sind. Dieses Zusammensetzen wird *Harmonisierung* genannt. Wir verwenden dazu folgende Methode aus [16, S. 34]. Grafisch wird diese Methode in Abbildung 2.1 dargestellt. Seien $(X^*(1), \dots, X^*(F^*))$ die extrahierten Merkmalsmatrizen der Merkmale $1, \dots, F^*$ und $W_e(i)$ die Größe des Extraktionsfensters des i -ten Merkmals. Als erstes wird das Minimum W_{min} aller Fenstergrößen berechnet. Dies ist dann die Fenstergröße der harmonisierten Matrix X^H . Für ein Stück der Länge l hat X^H also $\lfloor \frac{l}{W_{min}} \rfloor$ Spalten. Extraktionsfenster die größer als W_{min} sind, werden auf alle kleineren Fenster aufgeteilt, deren Anfang sie enthalten. Das letzte kleine Fenster, das den Wert des j -ten Fensters des i -ten Merkmals annimmt ist also $\lceil j \cdot \frac{W_e(i)}{W_{min}} \rceil$.

Nachdem die harmonisierte Matrix berechnet wurde, beginnt die eigentliche Merkmalsverarbeitung. Man kann dabei zwischen drei Methoden unterscheiden:

- *Vorverarbeitungsmethoden* sind Methoden, die die Dimensionen der Merkmalsmatrix nicht verändern, sondern nur deren Werte, wie z.B. fehlende Werte auffüllen oder die Werte normalisieren.
- Methoden, die die *Merkmalsdimension* verarbeiten, arbeiten auf den Reihen der Merkmalsmatrix. Häufig ist ihr Ziel die Matrix zu verkleinern, um die Geschwindigkeit und Qualität der Klassifikation zu verbessern. Dies kann z.B. durch Auswahl der wichtigsten Reihen der Matrix geschehen (Merkmalsauswahl).
- Methoden, die die *Zeitdimension* verarbeiten, arbeiten auf den Spalten der Matrix. Auch sie haben häufig die Aufgabe die Matrix zu verkleinern, indem Zeitabschnitte zusammengefasst werden, oder die musikalisch bedeutsame Zeitpunkte ausgewählt werden, während der Rest entfernt wird.

Im Folgenden stellen wir alle Merkmalverarbeitungsmethoden, die wir in dieser Arbeit verwenden, in der Reihenfolge vor, wie sie auf die Merkmale angewendet werden.

2.2.1 Ersetzung fehlender Werte

Bei der Extraktion von Merkmalen kann es oft passieren, dass für manche Zeitfenster kein Wert für eines der Merkmale berechnet werden kann. So sind z.B. viele Frequenzbereich-Features unmöglich zu berechnen, wenn das Signal zu leise ist [22, S. 371]. Daher sind Vorverarbeitungsmethoden wichtig, die fehlende Werte auf sinnvolle Weise auffüllen können, sodass auch Zeitfenster mit fehlenden Werten für die Klassifikation oder weitere Merkmalsverarbeitungen genutzt werden können.

Eine häufige Methode, mit fehlenden Werten umzugehen, ist es sie durch den Median des Merkmals zu ersetzen. Bestehe die Merkmalsmatrix X^H aus m Merkmalsdimensionen und n Zeitfenstern. Wenn für die Stelle $x_{u,w}$ der Matrix der Wert fehlt, wird also $x_{u,w} = \text{median}(\{x_{u,i} | 1 \leq i \leq n \text{ und der Wert von } x_{u,i} \text{ ist vorhanden}\})$ gesetzt. Diese Methode kommt auch in dieser Arbeit zum Einsatz und wird auf alle Merkmale angewendet.

2.2.2 Onset-Pruning

Nachdem man die Merkmale für die Musikstücke, die man untersuchen möchte, extrahiert hat, hat man eine sehr große Datenmenge. Schließlich hat man viele verschiedene Merkmale für eine große Menge von Musikstücken berechnet. Die meisten Merkmale werden für Zeitfenster berechnet, die nur ein Bruchteil einer Sekunde groß sind. Für ein einzelnes Stück gibt es also eine sehr große Menge von Merkmalsvektoren. Bei einer so großen Datenmenge kann das Klassifikationstraining und die Klassifikation selbst lange Zeit dauern. Eine Möglichkeit, die zeitliche Dimension der Merkmalsmatrix zu verringern, ist das *Onset-Pruning*. Dabei werden nur bestimmte dieser Zeitfenster behalten und die restlichen aussortiert. Es handelt sich also um eine Methode, die die Merkmalsdimensionalität der Merkmalsmatrix verringert. Um zu verstehen, wie dies funktioniert, müssen wir zunächst erklären, was Onsets sind.

Als Onset bezeichnet man den Beginn eines Tons. Man kann dabei zwischen drei Arten von Onsets unterscheiden: Dem *physikalischen Onset*, also dem Zeitpunkt, an dem die Lautstärke des Tons beginnt größer als 0 zu werden, dem *empfundenen Onset*, also dem Zeitpunkt, an dem ein Mensch den Beginn des Tons erstmals wahrnimmt und dem *empfundenen rhythmischen Onset*, womit man den Zeitpunkt, an dem ein Mensch den Ton rhythmisch wahrnehmen kann [22, S. 412]. Um einen Onset zu erkennen, müssen viele verschiedene Merkmale berücksichtigt werden. So kann man zum Beispiel die Onsets eines Klaviers recht leicht an der Lautstärke erkennen, da ein Ton kurz nach dem Anschlag durch den Hammer am lautesten ist und danach nur noch leiser werden kann [22, S. 413]. Bei vielen anderen Instrumenten, wie zum Beispiel Blasinstrumenten, kann sich die Lautstärke eines Tons allerdings beliebig während des Tons verändern [22, S. 413]. Dort muss man zusätzlich versuchen die Geräusche zu erkennen, die beim Beginn eines Tons oder

dem Übergang zwischen zwei Tönen entsteht. Bei polyphonen Aufnahmen, in denen viele verschiedene Instrumente spielen, ist dies natürlich besonders schwierig.

Beim Onset-Pruning werden, Zeitfenster ausgewählt, die in einer bestimmten Relation zu den erkannten Onsets stehen, in der Hoffnung, dass diese musikalisch besonders wichtig sind. Für gewöhnlich wählt man Zeitfenster, die einen Onset enthalten, oder, die den Zeitpunkt genau zwischen zwei Onsets enthalten. In [19] wurden verschiedene Methoden zur Reduzierung der zeitlichen Dimensionalität für die Klassifikation von Musik getestet. Besonders gut schnitt dabei die Methode ab, die Merkmalsmatrix nur auf Zeitfenster zu reduzieren, die die Mitte zwischen zwei Onsets enthalten. Die musikalische Begründung, woran dies liegen könnte, ist, dass sich der Klang zwischen zwei Onsets nur wenig ändert, da schließlich keine neuen Töne beginnen. Während und um einen Onset herum ist der Klang allerdings recht instabil. Der beste Zeitpunkt, den man zur Klassifikation auswählen kann, ist also die Mitte zwischen zwei Onsets. Diese Methode verwenden wir auch in dieser Arbeit. Wir wenden sie auf alle Merkmale außer den Merkmalsgruppen Akkorde und Instrumente an, da diese ohnehin nur auf Zeitfenstern der Größe 10s berechnet werden.

2.2.3 Normalisierung

Für viele Merkmalsverarbeitungs- und Klassifikationsmethoden ist es wichtig, dass die Werte der Merkmalsmatrix in einem bestimmten Wertebereich liegen, damit die Methoden richtig funktionieren. Für die strukturelle Komplexität (s. Abschnitt 2.2.4) müssen die Merkmale z.B. ausschließlich nicht negative Werte haben, da sie sonst nicht berechnet werden kann. Sowohl für die strukturelle Komplexität als auch für FKNN (s. Abschnitt 2.3.2) ist es außerdem wichtig, dass alle Merkmale im gleichen Wertebereich liegen, da sonst Merkmale mit einem größeren Wertebereich fälschlicherweise als wichtiger eingeschätzt werden. Die Werte der Merkmale auf einen vorgeschriebenen Wertebereich $[N_{min}, N_{max}]$ abzubilden, nennt man *Normalisierung* [22, S. 368]. In diesem Abschnitt stellen wir einige Methoden dazu vor.

Eine der einfachsten Methoden den Wertebereich eines Merkmals auf den Bereich $[N_{min}, N_{max}]$ abzubilden, ist die *Min-Max-Normalisierung*. Bei ihr werden die Werte eines Features linear auf den Bereich $[N_{min}, N_{max}]$ abgebildet, sodass sie sich nach der Normalisierung an der gleichen Stelle relativ zu $[N_{min}, N_{max}]$ befinden, wie sie vorher zu dem alten Minimum- und Maximumwert lagen.

Sei $x_{u,w}$ der u -te Wert des Merkmalsvektors des w -ten Zeitfensters. Sei $\min(x_u)$ das Minimum und $\max(x_u)$ das Maximum, das der u -te Wert der Merkmalsvektoren annehmen kann. Der normalisierte Wert $x'_{u,w}$ wird dann auf folgende Weise berechnet [22, S. 369]:

$$x'_{u,w} = \frac{x_{u,w} - \min(x_u)}{\max(x_u) - \min(x_u)} \cdot (N_{max} - N_{min}) + N_{min} \quad (2.1)$$

Häufig wählt man als $[N_{min}, N_{max}]$ das Intervall $[0, 1]$, so auch in dieser Arbeit. Der Vorteil der Min-Max-Normalisierung ist es, dass sie recht simpel und gut nachvollziehbar funktioniert. Die einzige Schwierigkeit ist es sinnvolle Werte für $min(x_u)$ und $max(x_u)$ zu finden. Wenn in den Daten, die man normalisieren möchte, Werte vorkommen, die kleiner als $min(x_u)$ oder größer als $max(x_u)$ sind, liefert die Normalisierung Werte, die außerhalb von $[N_{min}, N_{max}]$ liegen, was dazu führen kann, dass einige Algorithmen nicht mehr funktionieren, wie zum Beispiel die strukturelle Komplexität, die ein Abstandsmaß verwendet, das nur mit nichtnegativen Werten funktioniert. Deshalb ist es wichtig, diese Werte anhand einer großen Menge von Beispieldaten zu bestimmen. Einfach nur das konkrete Minimum und Maximum der zu normalisierenden Daten zu bestimmen, reicht dabei nicht aus, denn dann müsste man jedes mal, wenn man neue unbekannte Stücke klassifizieren möchte, möglicherweise die Werte anpassen, alle Merkmale neu verarbeiten und den Klassifizierungsalgorithmus neu trainieren. Es ist also wichtig, Werte zu finden, die für die meisten Daten funktionieren ohne sie zu ändern. In dieser Arbeit werden Werte verwendet, die anhand einer großen Menge von Musikstücken berechnet wurden [16, S. 37].

Eine Alternative zur Min-Max-Normalisierung, welche beliebige reelle Zahlen auf das Intervall $[0, 1]$ abbildet, ist die *Softmax-Normalisierung* [22, S. 369]. Sie hat den Vorteil, dass alle reellen Zahlen auf den gewünschten Wertebereich abgebildet werden. Sie funktioniert allerdings nur dann gut, wenn man möglichst genaue Werte für den Mittelwert und die Standardabweichung der zu normalisierenden Werte hat [22, S. 369–370]. Deshalb verwenden wir in dieser Arbeit die Softmax-Normalisierung nicht. Denn im Gegensatz zum Mittelwert und der Standardabweichung stehen uns Minimum- und Maximumwerte zur Verfügung, die aus einer großen Menge von Stücken berechnet wurden [16, S. 37]. Wir wenden die Normalisierung auf alle Merkmale außer den Merkmalsgruppen Chroma, Instrumente und Akkordvektor an, da diese bereits normalisiert sind.

2.2.4 Strukturelle Komplexität

Die *strukturelle Komplexität* ist eine Merkmalverarbeitungsmethode, die die Änderung eines Merkmals auf verschiedenen Zeitskalen berechnet. Sie wurde von Mauch und Levy in [12] vorgestellt. Mauch und Levy vermuten, dass strukturelle Komplexitätsmerkmale musikalische Komplexität beschreiben können [12].

Für ein Musikstück seien für N Zeitfenster die M -dimensionalen Merkmalsvektoren $x_1, x_2, \dots, x_N$ extrahiert worden. Diese N Merkmalsvektoren $x_1, \dots, x_N \in \mathbb{R}^M$ können zu einem strukturellen Komplexitätsmerkmal verarbeitet werden. Um die Komplexität des Ursprungsmerkmals für das Fenster i zu berechnen, werden die k Merkmalsvektoren $x_{i-k+1}, \dots, x_i \in \mathbb{R}^M$ vor i mit den k Merkmalsvektoren $x_{i+1}, \dots, x_{i+k} \in \mathbb{R}^M$ dahinter verglichen. Dazu wird eine Zusammenfassungsfunktion $s : (\mathbb{R}^M)^k \rightarrow \mathbb{R}^M$ verwendet, um die k Vektoren vor und die k Vektoren hinter dem zu untersuchenden Zeitpunkt zusammen-

zufassen, und ein Abstandsmaß $d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^+$, um den Abstand zwischen der linken Zusammenfassung $s_{[i-k+1:i]}$ und der rechten $s_{[i+1:i+k]}$ zu berechnen.

Diese Abstände werden für n verschiedene Fenstergrößen W_j mit $j = 1, \dots, n$ berechnet, die beschreiben, wie groß die Zeitfenster vor und nach dem aktuellen Zeitpunkt sind. Dadurch entsteht ein n -dimensionales Merkmal $v_i = (v_i^1, \dots, v_i^n)$, wobei v_i^j die Änderung am i -ten Fenster für die j -te Fenstergröße beschreibt und folgendermaßen berechnet wird [12]:

$$v_i^j = \begin{cases} d(s_{[i-W_j+1:i]}, s_{[i+1:i+W_j]}), & \text{falls } w_j \leq i \leq N - W_j \\ 0 & \text{sonst.} \end{cases} \quad (2.2)$$

Für die Fenster vor W_j und nach $N - W_j$ kann die Komplexität nicht berechnet werden, da dort nicht genügend Fenster vor bzw. nach dem Zeitpunkt vorhanden sind.

Als Fenstergrößen W_j kann man beliebige Werte verwenden. Mauch und Levy beschränken sich dabei auf Zweierpotenzen: $W_j = 2^{j-1}$. Wir nutzen ebenfalls Zweierpotenzen. Wir verwenden die Fenstergrößen 2s, 4s, 8s, 16s, 32s und 64s.

Als Zusammenfassungsfunktion verwenden Mauch und Levy den Mittelwert und als Abstandsmaß den *Jenson-Shannon* Abstand. Der Jenson-Shannon Abstand zweier Vektoren $s_1, s_2 \in \mathbb{R}^M$ wird auf folgende Weise berechnet [12]:

$$\frac{\mathbf{KL}(s_1||M) + \mathbf{KL}(s_2||M)}{2} \quad (2.3)$$

Dabei ist $M = \frac{s_1+s_2}{2}$ und $\mathbf{KL}$ der Kullback-Leibler Abstand [12]:

$$\mathbf{KL}(x||y) = \sum_{i=1}^n x_i \log \frac{x_i}{y_i} \quad (2.4)$$

In dieser Arbeit benutzen wir ebenfalls den Mittelwert und den Jenson-Shannon Abstand.

Da der Jenson-Shannon Abstand nur für positive Werte berechnet werden kann, muss das Ursprungsmerkmal normalisiert werden. Außerdem ermöglicht Normalisierung es, die Komplexität verschiedener Merkmale besser zu vergleichen, auch wenn sie ursprünglich Werte in sehr unterschiedlichen Größenordnungen hatten. Mauch und Levy normalisieren die Vektoren nach der Zusammenfassung [12]. Wir normalisieren sie hingegen vorher. Zur Normalisierung verwenden wir die Min-Max-Normalisierung (s. Abschnitt 2.2.3). Da wir Min-Max-Normalisierung zur Normalisierung und den Mittelwert für die Zusammenfassung verwenden, hat die Reihenfolge an dieser Stelle allerdings keine Auswirkung auf das Ergebnis.

2.2.5 Vorbereitung von Klassifikationsinstanzen

Die letzte Aufgabe der Merkmalsverarbeitung ist es, die Klassifikationsinstanzen vorzubereiten, auf denen der Klassifikationsalgorithmus tatsächlich arbeitet. Dafür müssen aus

der Merkmalsmatrix Vektoren erzeugt werden, indem man die Merkmale aggregiert. Die einfachste Methode ist es, über den Zeitraum des gesamten Stückes die Merkmale zu aggregieren, indem man verschiedene Statistiken wie Mittelwerte, Standardabweichungen oder ähnliches der Merkmale berechnet. Auf diese Weise wird die Merkmalsmatrix des gesamten Musikstücks zu einem Vektor zusammengefasst, die ein Klassifikationsalgorithmus verwenden kann. Normalerweise funktioniert die Methode allerdings nicht besonders gut. Dadurch, dass man über das gesamte Stück aggregiert, gehen viele Informationen verloren. Aus diesem Grunde teilt man das Stück für gewöhnlich in Partitionen auf, die sich z.B. zur Hälfte überlappen. Die Merkmale werden dann über jede dieser Partitionen aggregiert. Dadurch ergibt sich für jede Partition ein Vektor, der als Klassifikationsinstanz verwendet wird. Der Klassifikationsalgorithmus arbeitet dann also nicht auf ganzen Stücken, sondern nur auf diesen Teilen. Die Wahl der Größe dieser Partitionen kann schwierig sein. Man muss sie groß genug wählen, um wichtige musikalische Eigenschaften erkennen zu können, andererseits sollten sie nicht so groß sein, dass unterschiedliche musikalische Abschnitte, wie z.B. Intro, Strophe oder Bridge, vermischt werden [16, S. 34]. In [20] wurden verschiedene Größen für diese Partitionen getestet. Dabei hat sich eine Größe von 24s bei leichten Klassifikationsproblemen und eine Größe zwischen 4s und 5s bei schwereren Problemen als am besten gezeigt.

Zur Aggregation der Merkmale berechnen wir zunächst die strukturelle Komplexität auf den Partitionen (s. Abschnitt 2.2.4). Darauf berechnen wir dann Mittelwert, Median, Minimum, Maximum, oberes und unteres Quartil und die Varianz. So entsteht für jede Fenstergröße der strukturellen Komplexität ein sieben-dimensionales Merkmal. Die strukturelle Komplexität berechnen wir für jede der acht Merkmalsgruppen, welche in Abschnitt 2.1 beschrieben werden, einzeln. Auf diese Weise können die Merkmale die Komplexität verschiedener musikalischer Eigenschaften beschreiben. Da bei der Berechnung der strukturellen Komplexität die Merkmale über Zeiträume von mehreren Sekunden aggregiert werden, konnten wir keine kleinen Partitionsgrößen von ca. 5s verwenden, wie sie in [20] erfolgreich waren. Stattdessen verwenden wir soweit es möglich ist Partitionsgrößen von 24s, die sich um 12s überlappen. Bei einigen Einstellungen der strukturellen Komplexität mussten wir allerdings noch größere Partitionsgrößen verwenden. Wir richten uns dabei nach dem Vorgehen in [16]. Für die Fenstergrößen 2s, 4s, und 8s verwenden wir Partitionen der Größe 24s, die sich um jeweils 12s überlappen, für die Fenstergrößen 16s und 32s verwenden wir Partitionen der Größe 96s, die sich um 48s überlappen und für die Fenstergröße 32s verwenden wir Partitionen der Größe 138s, die sich um 69s überlappen.

2.2.6 Merkmalsauswahl

Häufig ist es schwierig zu wissen, welche Merkmale sich am besten für ein bestimmtes Problem eignen. Deshalb kann man dazu neigen, eine große Menge von Merkmalen zu

verwenden und zu hoffen, dass einige wichtige Merkmale dabei sind, und auch als solche von dem Klassifikationsalgorithmus erkannt werden. Zu viele Merkmale führen allerdings dazu, dass Training und Klassifikation sehr lange dauern. Außerdem kann sich die Qualität der Klassifikation sogar verschlechtern, wenn der Klassifikationsalgorithmus beim Training Eigenschaften der Musik lernt, die für das Problem eigentlich unwichtig sind. Dies führt zu *Overfitting* [22, S. 389], also zu einem Modell, dass sehr gut auf der Trainingsmenge, aber schlecht auf unbekannten Stücken funktioniert. Besonders bei Algorithmen wie FKNN (s. Abschnitt 2.3.2), die gar kein Training im eigentlichen Sinne haben, sondern einfach alle Merkmale gleich behandeln, können Merkmale, die für die Klassifikation unwichtig sind, schaden. Aus diesem Grunde kann es sehr wichtig sein, die Menge der verwendeten Merkmale automatisch auf die wichtigsten zu reduzieren.

Eine Methode zur Merkmalsauswahl, die in [6, S. 98–99] beschrieben wird, ist *Relief*. Sie gehört zu den *Filtermethoden*. Das heißt, welche Merkmale ausgewählt werden, wird direkt anhand der Daten bestimmt, ohne Informationen darüber zu nutzen, wie gut der Klassifikator tatsächlich mit den ausgewählten Merkmalen funktioniert [6, S. 89].

Zum Filtern wird für jede Merkmalsdimension die *Relief-Relevanz* berechnet, die beschreiben soll, wie wichtig diese Dimension für das Klassifikationsproblem ist. Alle Merkmalsdimensionen, deren Relevanz über einem bestimmten Wert liegen, werden dann für die Klassifikation verwendet.

Die Grundidee der Relief-Merkmalsauswahl, ist es die Merkmale auszuwählen, bei denen die Werte der Instanzen unterschiedlicher Klassen möglichst weit auseinander liegen. Sie arbeitet dabei auf den bereits zusammengefassten Klassifikationsinstanzen (s. Abschnitt 2.2.5). Es wird dann für jede Merkmalsdimension und jede Klassifikationsinstanz der Abstand dieser Instanz zu den K nächsten Instanzen, die der gleichen Klasse angehören, und den K nächsten Instanzen, die einer anderen Klasse angehören berechnet. Und dann wird für jede Merkmalsdimension das Verhältnis dieser beiden Abstände berechnet. Dies ist dann die Relevanz dieser Merkmalsdimension. Auf diese Weise erhalten Merkmalsdimensionen mit großen Abständen zwischen unterschiedlichen Klassen und kleinen Abständen in einer Klasse untereinander eine große Relevanz. Sei n die Anzahl aller Klassifikationsinstanzen. Sei K die Anzahl der nächsten Nachbarn, die zur Berechnung verwendet werden sollen. Sei $x_{u,j}$ der Wert der u -ten Merkmalsdimension der j -ten Klassifikationsinstanz. Sei $\{x_{(\text{nearest-hit})_{k,j}} | 1 \leq k \leq K\}$, die Menge der nächsten Nachbarn, die der gleichen Klasse angehören, und $\{x_{(\text{nearest-miss})_{k,j}} | 1 \leq k \leq K\}$, die Menge der nächsten Nachbarn, die einer anderen Klasse angehören. Die nächsten Nachbarn werden dabei

mithilfe des euklidischen Abstands berechnet. Die Relief-Varianz der u -ten Merkmalsdimension berechnet sich dann folgendermaßen:

$$R_{\text{RELIEF}_K}(x_u) = \frac{\sum_{j=1}^n \sum_{k=1}^K \|x_{u,j} - x_{(\text{nearest-miss})_{kj}}\|}{\sum_{j=1}^n \sum_{k=1}^K \|x_{u,j} - x_{(\text{nearest-hit})_{kj}}\|} \quad (2.5)$$

Danach werden alle Merkmalsdimensionen, deren Relief-Relevanz größer als ein Wert $\tau \in \mathbb{R}$ ist, behalten und der Rest entfernt. Man sollte einen Wert für τ wählen, der mindestens 1 ist, da Werte kleiner 1 bedeuten, dass die falschen Nachbarn bei diesem Merkmal näher liegen als die richtigen. Wir haben $\tau = 1, 1$ gewählt.

Der Vorteil der Relief-Merkmalsauswahl ist, dass sie im Vergleich zu anderen Methoden, wie z.B. evolutionärer Merkmalsauswahl, wie sie z.B. in [16] zum Einsatz kommt, leicht zu implementieren und weniger rechenaufwändig ist, da sie nur einmal durchgeführt werden muss. Außerdem passt diese Methode sehr gut zu dem Algorithmus FKNN, den wir in dieser Arbeit verwenden, da auch FKNN die nächsten Nachbarn einer Instanz verwendet, um diese zu klassifizieren (s. Abschnitt 2.3.2). Daher ist es sinnvoll, dass Merkmale so gewählt werden, dass Nachbarn mit der gleichen Klasse möglichst nah beieinander liegen. Ein Nachteil der Relief-Merkmalsauswahl ist allerdings, dass dadurch, dass nur einmal gefiltert wird, die tatsächliche Klassifikationsqualität nicht berücksichtigt wird und es so keine Garantie gibt, dass sie die Qualität der Klassifikation tatsächlich verbessert.

2.3 Klassifikation

Im folgenden Abschnitt wird beschrieben, was ein Klassifikationsproblem ist. Außerdem erklären wir, wie man Musikgenreklassifikation als Klassifikationsproblem verstehen kann, und stellen den Algorithmus vor, den wir in dieser Arbeit zur Klassifikation verwenden.

2.3.1 Überwachte Klassifikation

Bei einem Klassifikationsproblem sollen Instanzen in Klassen klassifiziert werden. Einen Klassifikator, der dies tut, kann man als Funktion $f : X \rightarrow Y$ verstehen [22, S. 303]. Dabei ist X die Menge aller möglichen zu klassifizierenden Instanzen und Y die Menge der Klassen. Beispielsweise könnte X die Menge aller möglichen Musikstücke sein und Y eine Menge von Genres, in die man die Musik einteilen möchte. Dies ist bei der *Musikgenreklassifikation* der Fall, welche wir in dieser Arbeit untersuchen. Ziel ist es dabei, die Funktion f anhand von Beispielen zu lernen. Sei $\{x_1, \dots, x_n\} \subset X$ eine Menge von Trainingsbeispielen und sei jedem x_i ein $y_i \in Y$ zugeordnet. Die Menge $\{(x_1, y_1), \dots, (x_n, y_n)\}$ der Paare von Instanzen und den zugehörigen Klassen nennen wir *Trainingsmenge*. Die Paare von je einer Instanz und der dazugehörigen Klasse nennen wir *beschriftete* Instanzen.

Diese Beschriftung muss vorher von einem Experten vorgenommen werden, der die Klassen kennt und die Klassifikation überwacht. Aus diesem Grund nennt man diese Art der Klassifikation auch *überwachte Klassifikation*. Aufgabe eines *Klassifikationsalgorithmus* ist es nun anhand dieser Trainingsmenge eine Funktion $f : X \rightarrow Y$ zu finden, die Instanzen klassifizieren kann. Diese Funktion nennen wir *Modell*. Ziel ist es ein Modell zu finden, dass möglichst wenige Fehler macht, auch wenn es unbekannte Instanzen klassifiziert, die nicht in der Trainingsmenge enthalten waren. Wenn für ein $x \in X$, das der Klasse $y \in Y$ angehört, $f(x) \neq y$ ist, nennen wir das einen Fehler.

Eine Besonderheit von Musikklassifikation ist, dass man häufig nicht ganze Musikstücke klassifiziert, sondern die zu klassifizierenden Stücke zunächst in Partitionen einteilt, die einige Sekunden lang sind und sich um einige Sekunden überlappen. Diese Partitionen sind dann die Instanzen, mit denen trainiert und klassifiziert wird. Am Ende wird ein Musikstück dann der Klasse zugeordnet, der die Mehrheit seiner Partitionen zugeordnet worden sind. Wie diese Einteilung in Partitionen genau funktioniert, wird in Abschnitt 2.2.5 beschrieben.

Wir verwenden in dieser Arbeit *binäre Klassifikation* [22, S. 305], das heißt anstatt ein Modell $f : X \rightarrow Y$ zu trainieren, dass alle Klassen klassifizieren kann, trainieren wir für jede Klasse $Y_i \in Y$ ein einzelnes Modell $f_i : X \rightarrow \{0, 1\}$, das bestimmt, ob eine Instanz der i -ten Klasse angehört oder nicht. Mit den Modellen $f_1, \dots, f_m$ kann man dann alle Klassen $Y = \{Y_1, \dots, Y_m\}$ klassifizieren. Der Klassifikationsalgorithmus FKNN, den wir in dieser Arbeit verwenden, (s. Abschnitt 2.3.2) ist ein *Fuzzy*-Klassifikator, das heißt anstatt Modelle $f_1, \dots, f_m : X \rightarrow \{0, 1\}$ zu berechnen, die bestimmen, ob eine Instanz einer Klasse angehört oder nicht, berechnet er Modelle $f'_1, \dots, f'_m : X \rightarrow [0, 1]$, die einen zwischen 0 und 1 liegenden Grad der Zugehörigkeit zu einer Klasse bestimmen.

2.3.2 Der Fuzzy- k -nearest-Neighbors-Algorithmus

In dieser Arbeit wird zur Klassifikation der Algorithmus *Fuzzy- k -nearest-Neighbors* (kurz FKNN) benutzt, wie ihn Keller, Gray und Givens vorgestellt haben [8]. Im folgenden Abschnitt werden wir ihn näher beschreiben.

Der Algorithmus ist eine Weiterentwicklung des Algorithmus *k -Nearest-Neighbors* (kurz KNN). Bei KNN werden für eine zu klassifizierende Instanz x die k nächsten Nachbarn aus den Trainingsinstanzen berechnet. Dazu wird der euklidische Abstand der Merkmalsvektoren verwendet. Das zu klassifizierende x wird dann der Klasse zugeordnet, der die Mehrheit der k nächsten Nachbarn angehören. Der Nachteil dieser simplen Mehrheitsentscheidung ist, dass alle k Nachbarn die gleiche Auswirkung auf das Ergebnis haben, auch, wenn sie sehr unterschiedliche Entfernungen zu x haben können.

Die Idee hinter FKNN ist es, die Auswirkung der einzelnen Nachbarn anhand des Abstandes zu x zu gewichten. Auf diese Weise haben die näheren Nachbarn eine größere

Auswirkung auf die Klassifikation. Dies ist sinnvoll, da man davon ausgeht, dass näher beieinander liegende Instanzen auch mit höherer Wahrscheinlichkeit der gleichen Klasse angehören. Dies ist schließlich bereits die Grundidee hinter dem normalen KNN-Algorithmus. Auf diese Weise wird x nicht einfach fest in die Klasse der Mehrheit klassifiziert, sondern erhält zu jeder Klasse einen Zugehörigkeitsgrad, der im Intervall $[0, 1]$ liegt. Auch die Trainingsinstanzen müssen nicht jeweils fest einer der Klassen zugeordnet sein, sondern besitzen zu jeder Klasse einen Zugehörigkeitsgrad. Diese Eigenschaft, dass die Instanzen nicht nur zu einer Klasse gehören können oder nicht, sondern zu einem gewissen Grad zu dieser Klasse gehören, ist die Grundidee von *Fuzzy-Mengen*, die 1965 von Zadeh in [24] eingeführt wurden. Sie bilden die Grundlage der *Fuzzy-Logik*, einer unscharfen Erweiterung der booleschen Logik, die es ermöglicht ungenaue und unsichere Information darzustellen. Auf dieser Grundlage wurden viele Methoden und Algorithmen entwickelt, die mit solchen Zugehörigkeitsgraden arbeiten. Der FKNN-Algorithmus wurde als fuzzy Variante des gewöhnlichen KNN-Algorithmus entwickelt. Er wird in [8] beschrieben und mit dem normalen KNN-Algorithmus verglichen. Er erreichte dabei bessere Ergebnisse als der gewöhnliche KNN Algorithmus. Außerdem schienen die berechneten Zugehörigkeitsgrade auf sinnvolle Weise die Sicherheit der Klassifikation wiederzugeben.

Sei $L = \{x_1, x_2, \dots, x_n\}$ die Menge der Trainingsbeispiele und $Y = \{y_1, \dots, y_m\}$ die Klassen, in die klassifiziert werden soll. Sei r_{ij} der Zugehörigkeitsgrad von x_j zur Klasse y_i , welcher bereits bekannt ist. Sei x die Instanz die klassifiziert werden soll und $r_i(x)$, der Zugehörigkeitsgrad von x zur Klasse y_i , welcher bestimmt werden soll, und k die Anzahl der nächsten Nachbarn von x , die berücksichtigt werden sollen. Der FKNN-Algorithmus funktioniert dann folgendermaßen:

Eingabe: $L, r_{1,1}, \dots, r_{mn}$

Ausgabe: $r_1(x), \dots, r_m(x)$

$i \leftarrow 1$

repeat

Berechne Abstand zwischen x und x_i .

if $i \leq k$ **then**

Füge x_i zur Menge der k nächsten Nachbarn hinzu.

else if x_i näher an x liegt als einer der vorherigen nächsten Nachbarn **then**

Entferne den am weitesten entfernten Nachbarn und füge x_i zur Menge der k nächsten Nachbarn hinzu.

end if

$i \leftarrow i + 1$

until $i = n$

$i \leftarrow 1$

repeat

Berechne $r_i(x)$ mithilfe von Gleichung 2.6.

$i \leftarrow i + 1$

until $i = m$

Algorithmus 2.1: FKNN, nach [8]

$$r_i(x) = \frac{\sum_{j=1}^k r_{ij} \left(\frac{1}{\|x - x_j\|} \right)^{\frac{2}{(\mu-1)}}}{\sum_{j=1}^k \left(\frac{1}{\|x - x_j\|} \right)^{\frac{2}{(\mu-1)}}} \quad (2.6)$$

In der Gleichung 2.6 [8] sind $\{x_1, \dots, x_k\}$ die k nächsten Nachbarn von x . Die Variable μ gibt an wie stark der Einfluss der Nachbarn abhängig von ihrer Entfernung abnimmt. Bei $\mu = 2$ wird der Einfluss eines Nachbarn mit dem Kehrwert des quadratischen Abstands zu der zu klassifizierenden Instanz gewichtet. Je höher μ ist, desto langsamer nimmt die Gewichtung mit dem Abstand ab und umso mehr können auch noch weit entfernte Nachbarn Einfluss nehmen. Wir verwenden in dieser Arbeit den Wert $\mu = 2$, so wie es auch Keller, Gray und Givens in [8] getan haben.

Dass FKNN Zugehörigkeitsgrade statt festen Zuordnungen zurückgibt, hat den Vorteil, dass man nach der Klassifikation nicht nur weiß, welcher Klasse eine Instanz wahrscheinlich angehört, sondern auch ein Indiz dafür hat, wie sicher man sich bei dieser Klassifikation sein kann. Wenn für eine Instanz z.B. eine Zugehörigkeit von 0,9 zu einer Klasse und 0,05 zu zwei anderen Klassen berechnet wird, kann man sich relativ sicher sein, dass die Instanz zur ersten Klasse gehört. Wenn eine Zugehörigkeit von 0,55 zur ersten, 0,44 zur zweiten und 0,01 zur dritten berechnet wird, ist es schwierig zu sagen, ob die Instanz zu ersten oder

zweiten Klasse gehört. Aber man kann sich sicher sein, dass sie nicht zur dritten gehört [8]. Auf diese Weise lassen sich bei der Genreklassifikation auch gut Musikstücke modellieren, die sich nicht genau einem Genre zuordnen lassen. Da die Eingabe als auch die Ausgabe des FKNN unscharf sind, lässt er sich auch in größere Fuzzy-Systeme einbauen.

Da die Stücke in unseren Testdaten von Experten jeweils genau einem Genre zugeordnet wurden, sind die Zugehörigkeitsgrade unserer Trainingsdaten dennoch binär, also nur 0 oder 1. Die Ausgabe des FKNN ist trotzdem fuzzy. Zur Bewertung werden die Ergebnisse allerdings wieder gerundet, da die meisten Maße für binäre Klassifikation gedacht sind. Trotzdem wird es interessant sein zu sehen, wie sich die errechneten Zugehörigkeitsgrade einzelner Stücke auf die verschiedenen Genres verteilen.

2.4 Bewertung

Nachdem wir gesehen haben, wie die Klassifikation von Musikstücken funktioniert, erklären wir nun, wie man die Leistung eines Klassifikators bewerten kann.

2.4.1 Resampling

Wir haben gesehen, dass ein guter Klassifikationsalgorithmus in der Lage sein sollte, anhand der Trainingsmenge ein Modell zu finden, welches auch bei unbekannten Instanzen, die nicht in der Trainingsmenge enthalten waren, eine geringe Fehlerwahrscheinlichkeit hat. Um zu testen, ob das Modell, das ein Algorithmus berechnet hat, gut ist, muss man es also auf Daten testen, die in der Trainingsmenge nicht enthalten waren. Aus diesem Grunde trainiert man Algorithmen, die man untersuchen möchte, normalerweise nicht auf den gesamten Daten, die einem zu Verfügung stehen, sondern nur auf einem Teil. Auf diese Weise hat man nach dem Training immer noch Instanzen übrig, deren Klasse man kennt, und auf denen man den Algorithmus testen kann. Diese Aufteilung in Trainings- und Testmenge nennt man *Resampling*. Es gibt viele Resamplingmethoden. In dieser Arbeit verwenden wir die Methoden *Holdout* und *Kreuzvalidierung*. Diese Methoden werden im Folgenden beschrieben.

Holdout

Holdout, auch *Train and Test* genannt, ist die einfachste der Resamplingmethoden [22, S. 334]. Bei ihr wird einfach ein Teil der Daten zum Trainieren verwendet und auf dem Rest wird das daraus entstandene Modell getestet.

Sei $L = \{(x_1, y_1), \dots, (x_n, y_n)\}$ die Menge aller beschrifteten Instanzen, die uns zur Verfügung stehen. Beim Holdout wird L in eine Trainingsmenge L' und eine Testmenge T eingeteilt, sodass $L = L' \cup T$ gilt [22, S. 334]. Der Algorithmus wird dann auf L' trainiert und das daraus entstehende Modell wird auf T getestet und bewertet.

Diese Methode verwenden auch wir. Bei uns gilt allerdings nicht $L = L' \cup T$ sondern lediglich $L' \cup T \subset L$. Wir benutzen also nicht alle Daten zum Trainieren und Testen. Wir benutzen beim Training für jedes Genre eine eigene Trainingsmenge, in der gleich viele positive und negative Beispiele für dieses Genre enthalten sind. Zum Testen verwenden wir aber für jedes Genre die gleiche Testmenge. Die genaue Zusammensetzung der Trainings- und Testmengen wird im Abschnitt 3.1.1 beschrieben.

Kreuzvalidierung

Eine weitere Resampling-Methode, die wir verwenden, ist die *Kreuzvalidierung*. Bei dieser Methode wird die zur Verfügung stehende Datenmenge in $B \in \mathbb{N}$ gleichgroße Teile geteilt. Danach wird auf $B - 1$ dieser Teile trainiert und auf dem übrigen Teil getestet. Dies wird B mal wiederholt, sodass jeder der B Teile einmal die Testmenge war. Die Einteilung in die B Teile geschieht dabei zufällig.

Wenn $B = n$ gilt, nennt man das *leave-one-out-Kreuzvalidierung*, da man immer auf allen Beispielen außer einem trainiert und auf diesem übrigen Beispiel testet. Wir benutzen in dieser Arbeit zehnfache Kreuzvalidierung ($B = 10$). Wir teilen allerdings die Daten nicht einfach nur zufällig in zehn Teile auf, sondern achten dabei darauf, dass die Genres in jedem der zehn Teile mit ungefähr der gleichen Verteilung auftreten, wie in der ursprünglichen Menge. Das nennt man *stratifizierte Kreuzvalidierung* [9]. Dadurch wird verhindert, dass der Algorithmus auf einseitigen Trainingsmengen trainiert wird. Außerdem trainieren wir unseren Algorithmus nicht auf der gesamten Trainingsmenge, sondern nur auf einer Teilmenge, die gleich viele positive und negative Beispiele für das jeweilige Genre hat.

2.4.2 Bewertungsmaße

In den letzten Abschnitten wurde beschrieben, was ein Klassifikationsalgorithmus ist und wie man einen Datensatz in Trainings- und Testdaten einteilt, auf denen man ihn trainiert und testet. Nun stellt sich aber die Frage, wie man die Leistung eines Klassifikators auf den Testdaten bewertet. Wir stellen in diesem Abschnitt Maße vor, mit denen man dies messen kann.

Die Maße, die wir hier vorstellen, eignen sich nur für binäre Klassifikationen, also Klassifikationen, bei denen nur entschieden werden muss, ob eine Instanz zur fraglichen Klasse gehört oder nicht. Im Folgenden nennen wir Instanzen, die der Klasse angehören positiv und Instanzen, die der Klasse nicht angehören, negativ. Wir unterscheiden dabei zwischen vier Arten von Instanzen [16, S. 37]:

- Unter *wahren Positiven* verstehen wir die positiven Instanzen der Trainingsmenge, die auch als solche erkannt werden. Die Anzahl aller wahren Positiven bezeichnen

wir mit TP (true positives).

- Als *wahre Negative* bezeichnen wir alle negativen Instanzen der Trainingsmenge, die auch als solche erkannt werden. Die Anzahl aller wahren Negativen bezeichnen wir mit TN (true negatives).
- Als *falsche Positive* bezeichnen wir alle negativen Instanzen der Trainingsmenge, die fälschlicherweise als positiv klassifiziert wurden. Die Anzahl aller falschen Positiven bezeichnen wir mit FP (false positives).
- Unter *falschen Negativen* verstehen wir alle positiven Instanzen der Testmenge, die fälschlicherweise als negativ klassifiziert wurden. Die Anzahl dieser Instanzen bezeichnen wir mit FN (false negatives).

Das einfachste Maß, um die Qualität der Klassifikation eines Klassifikators zu beurteilen, ist der *relative Fehler* [16, S. 74]:

$$m_{RE} = \frac{FP + FN}{TP + TN + FP + FN} \quad (2.7)$$

Dies ist ein beliebtes Maß, da es sehr einfach zu interpretieren ist. Es gibt schließlich lediglich den Anteil der Klassifikationen an, die Fehler sind. Es hat jedoch vor Allem bei unausgeglichen Testmengen einen Nachteil. Angenommen, die Testmenge besteht zu 90% aus negativen Instanzen und zu 10% aus positiven Instanzen. Ein Klassifikator, der einfach jede Instanz als negativ klassifiziert, klassifiziert 90% der Instanzen korrekt und hat damit einen relativen Fehler von 0,1. Trotzdem ist er kein guter Klassifikator, da er kein Wissen über das tatsächliche Klassifikationsproblem besitzt. Aus diesem Grunde kann es sinnvoll, sein den *balancierten relativen Fehler* zu verwenden, der auf folgende Weise berechnet wird [16, S. 74]:

$$m_{BRE} = \frac{1}{2} \left(\frac{FN}{TP + FN} + \frac{FP}{TN + FP} \right) \quad (2.8)$$

Beim relativen balancierten Fehler wird also der relative Fehler der positiven und negativen Instanzen einzeln berechnet und diese beiden Werte werden dann gleichmäßig gewichtet. Wie man leicht sieht, gilt beim balancierten relativen Fehler, genau wie beim normalen relativen Fehler für einen Klassifikator, der stets richtig liegt, $m_{BRE} = 1$ und für einen Klassifikator, der immer falsch liegt $m_{BRE} = 0$. Interessant ist das Verhalten bei einem Zufallsklassifikator, der jede Instanz mit einer Wahrscheinlichkeit von $\alpha \in [0, 1]$ als positiv klassifiziert. Der Erwartungswert des balancierten relativen Fehlers für einen solchen Klassifikator ist immer $E(m_{BRE}) = \frac{1}{2}(\alpha + (1 - \alpha)) = \frac{1}{2}$. Das Beispiel des Klassifikators, der alle Instanzen als negativ klassifiziert, ist ein Spezialfall, bei dem $\alpha = 0$ gilt. Der

balancierte relative Fehler hat also in dieser Situation nicht das Problem, das der normale relative Fehler hat. Außerdem lässt sich der balancierte relative Fehler auch insofern gut interpretieren, als dass ein Wert, der kleiner als 0,5 ist, bedeutet, dass der Klassifikator vermutlich besser als ein Zufallsklassifikator ist.

In dieser Arbeit verwenden wir zur Auswertung den balancierten relativen Fehler, da wir zum Teil sehr unausgeglichene Testmengen haben, die zu einem Großteil aus negativen Beispielen bestehen.

Kapitel 3

Experimente

In diesem Kapitel werden die Experimente beschrieben, die im Rahmen dieser Arbeit durchgeführt werden, um die Merkmalsverarbeitung mit struktureller Komplexität (siehe Abschnitt 2.2.4) und die Klassifikation mit FKNN (siehe Abschnitt 2.3.2) zu untersuchen.

3.1 Versuchsaufbau

Um die Musikgenreklassifikation mit struktureller Komplexität und FKNN zu untersuchen, testen wir die beiden Algorithmen mit verschiedenen Einstellungen und bewerten die Ergebnisse.

Eine der wichtigsten Einstellungen ist natürlich, welche Merkmale zur Verarbeitung und Klassifikation verwendet werden. In dieser Arbeit verwenden wir die sieben Merkmalsgruppen, die Votolkin bereits in [16] benutzt hat, und in Abschnitt 2.1 näher beschrieben werden. Außerdem testen wir das Merkmal *Chord Vector*, welches die durch das Chordino Vamp Plugin [11] berechneten Akkorde als binären Vektor darstellt (s. Abschnitt 2.1). Eine weitere wichtige Einstellung für die strukturelle Komplexität sind die Fenstergrößen W , über die die Merkmale zusammengefasst werden (s. Abschnitt 2.2.4). In dieser Arbeit werden wir die Fenstergrößen 2s, 4s, 8s, 16s, 32s und 64s testen. Wir berechnen dabei die strukturelle Komplexität für jede Fenstergröße einzeln. Da die Merkmale der Merkmalsgruppen Akkorde und Instrumente auf Zeitfenstern der Größe 10s berechnet werden, kann für diese Merkmale die strukturelle Komplexität nur für die Fenstergrößen 16s, 32s und 64s berechnet werden. Die einzige Einstellung des FKNN, die wir variieren werden, ist die Anzahl k der Nachbarn, die einen Einfluss auf die Klassifikation haben. Wir untersuchen hier die Werte $k = 2, 4, 6, 8$.

Es gibt also acht Merkmalsgruppen, sechs Fenstergrößen und vier Werte für k die wir testen möchten. Dazu klassifizieren wir die verschiedenen Genres mit allen Kombinationen dieser Einstellungen und bewerten die Ergebnisse, indem wir den balancierten relativen Fehler berechnen, der in Abschnitt 2.4.2 näher beschrieben wird.

3.1.1 Testdaten

Für die Experimente werden die Musikdatenbank *CDs* des Lehrstuhls 11 der Fakultät Informatik der TU Dortmund und die Datenbank *1517* aus [4] verwendet.

CDs-Datenbank

Die CDs-Datenbank besteht aus 120 Alben, die den Genres *Classical*, *Electronic*, *Jazz*, *Pop*, *Rap* und *R'n'B* angehören und der Menge *TAS120*, die aus weiteren 120 Stücken besteht, die den gleichen Genres angehören. Die Stücke der 120 Alben sind in die Trainingsmenge *OS120* und die Testmenge *TS120* aufgeteilt. Die drei Mengen *OS120*, *TS120* und *TAS120* bestehen aus jeweils 120 Stücken. Jeweils 45 Stücke gehören dem Genre Pop/Rock an und jeweils 15 Stücke den anderen Genres. Die Musikstücke wurden von Musikexperten der Webseite AllMusicGuide¹ in diese Genres eingeteilt.

Für die Experimente wird für jedes Genre eine Trainingsmenge aus 30 Stücken der Menge *OS120* gebildet. Diese Trainingsmengen bestehen aus jeweils 15 positiven und 15 negativen Beispielen für das jeweilige Genre. FKNN wird für jedes Genre und jede Einstellung der Parameter einzeln auf der jeweiligen Trainingsmenge trainiert und dann auf den Testmengen *TS120* und *TAS120* getestet. Wir entfernen dazu vor der Aufteilung in Trainings- und Testmengen alle Stücke, die kürzer als 138s sind, damit die strukturelle Komplexität für alle Fenstergrößen berechnet werden kann. Bei Erstellung der Trainingsmengen erlauben wir allerdings für die positiven Beispiele auch kürzere Stücke, da für die meisten Genres nur 15 Stücke in *OS120* enthalten sind. Auf diese Weise können wir trotzdem Trainingsmengen mit 30 Stücken bilden. Für die größeren Fenstergrößen, können dann allerdings nicht immer alle positiven Beispiele verwendet werden. Von *TS120* und *TAS120* bleiben ohne die kurzen Stücke noch 235 Stücke übrig. Um die Ergebnisse zu bewerten wird der balancierte relative Fehler berechnet. Wir trainieren den Algorithmus auf den aus 30 Stücken bestehenden Mengen und nicht auf der gesamten Menge *OS120*, da FKNN nicht sehr gut funktioniert, wenn er auf unbalancierten Mengen trainiert. Außerdem lässt sich auf diese Weise gut erkennen, wie FKNN mithilfe von struktureller Komplexität von kleinen Trainingsmengen lernen kann, welche einem in der Praxis oft nur zur Verfügung stehen.

1517-Datenbank

Die Datenbank 1517 besteht aus 1674 Stücken, die in 23 Genres eingeteilt sind. Wir testen allerdings nur die Klassifikation der sechs Genres *Classical*, *Electronic and Dance*, *Jazz*, *Pop/Rock*, *Hip-Hop* und *R'n'B and Soul*, damit die Ergebnisse möglichst vergleichbar mit denen der CDs-Datenbank sind.

¹<http://www.allmusic.com/> (zuletzt geprüft am 23.09.2018 um 21:18 Uhr)

Vor unseren Experimenten entfernen wir aus der Datenbank alle Stücke, die kürzer als 138s sind, damit die strukturelle Komplexität für alle Fenstergrößen berechnet werden kann. Damit verbleiben noch 1550 Stücke in der Datenbank. Für unsere Experimente auf der 1517-Datenbank verwenden wir *stratifizierte Kreuzvalidierung*, wie sie von Kohavi in [9] beschrieben wird (siehe Abschnitt 2.4.1). Dazu wird die Menge so in Partitionen eingeteilt, dass die Genres in den Partitionen ungefähr die gleiche Verteilung haben, wie auf der gesamten Menge. Das Testen findet für jedes der sechs Genres einzeln statt. Aus den Trainingspartitionen wird dafür zufällig eine Trainingsmenge gezogen, die zur Hälfte aus positiven und zur Hälfte aus negativen Beispielen für das jeweilige Genre besteht, da FKNN besonders gut auf balancierten Trainingsmengen arbeitet. Auf dieser Menge wird FKNN trainiert und dann auf der Testpartition getestet.

3.1.2 Merkmalsauswahl mit Relief

Eine Möglichkeit die Klassifikationsergebnisse zu verbessern ist, vor dem Training die Menge der verwendeten Merkmale zu verringern. Dies ist besonders bei FKNN sinnvoll, da dieser Algorithmus alle Merkmale gleich stark gewichtet und nicht selbst lernt, welche Merkmale zur Klassifikation wichtig sind.

Wir verwenden dazu die *Relief-Relevanz*, die in [6, S. 98–99] vorgestellt wird und in Abschnitt 2.2.6 näher beschrieben wird. Die Relief-Varianz ist ein Maß, das sich aus dem Abstand der Trainingsbeispiele zu den K nächsten positiven und den K nächsten negativen Nachbarn berechnet und beschreibt, wie stark der Zusammenhang eines Merkmals und der zu klassifizierenden Klasse ist. Wir berechnen für jedes Genre die Relevanz aller möglichen Merkmale die zur Klassifikation verwendet werden können. Von diesen Merkmalen werden dann die mit der höchsten Relevanz für Training und Klassifikation ausgewählt. Um die Relief-Merkmalsauswahl anzuwenden, setzen wir die unterschiedlich verarbeiteten Merkmale wieder zusammen, ähnlich, wie die Methode zum Harmonisieren der Merkmalsmatrix, die in Abschnitt 2.2 beschrieben wird. Da die Partitionsgrößen der verschiedenen verarbeiteten Merkmale unterschiedlich sind, müssen die großen Partitionen auf die kleineren aufgeteilt werden, sodass Partitionen der Größe 24s entstehen, die sich um jeweils 12s überlappen und alle verarbeiteten Merkmale enthalten, die zu diesem Zeitpunkt des Stücks gehören. Für jede Dimension dieser Merkmalsmenge berechnen wir dann die Relief-Varianz auf der Trainingsmenge mit $K = 8$ und wählen alle Merkmale mit einer Relevanz von über $\tau = 1,1$ aus. Die ausgewählten Merkmale werden dann zur Klassifikation auf der Testmenge verwendet.

Anhand der Relief-Relevanz der Merkmale für die verschiedenen Genres können wir erkennen, welche Merkmale sich vermutlich gut zur Klassifikation für die von uns untersuchten Genres eignen. Um zu entscheiden, ob die Relief-Merkmalsauswahl für diese

Klassifikationsprobleme sinnvoll ist vergleichen wir den balancierten relativen Fehler der Klassifikation mit der vollständigen Merkmalsmenge und der Relief-Auswahl.

3.2 Ergebnisse

Nachdem wir im letzten Abschnitt den Aufbau der Versuche beschrieben haben, die wir im Rahmen dieser Arbeit durchführen, präsentieren wir nun die Ergebnisse. Wir haben die Klassifikation für alle im letzten Abschnitt beschriebenen Einstellungen durchgeführt und den balancierten relativen Fehler berechnet. Die Fehler für alle diese Klassifikationen kann man im Anhang A.2 finden. In den folgenden Abschnitten untersuchen wir diese Ergebnisse darauf, wie gut die in dieser Arbeit untersuchten Methoden für die unterschiedlichen Einstellungen funktionieren.

3.2.1 Strukturelle Komplexität

Im diesem Abschnitt werten wir die Ergebnisse der Experimente aus, um herauszufinden, welche Einstellungen der strukturellen Komplexität besonders gut funktionieren. Wir untersuchen dabei, welche Merkmalsgruppen und welche Fenstergrößen sich zur Klassifikation der verschiedenen Genres eignen.

Merkmalsgruppen

Unsere erste Fragestellung zum Verfahren der strukturellen Komplexität ist, welche Merkmalsgruppen sich gut zur Klassifikation eignen. Wir wollen dabei untersuchen, welche Merkmalsgruppen sich allgemein gut oder weniger gut eignen und welche Merkmalsgruppen sich besonders gut für bestimmte Genres eignen. Dazu berechnen wir für jede Kombination aus Genre und Merkmalsgruppe den durchschnittlichen balancierten relativen Fehler. Wir berechnen dazu den Mittelwert des Fehlers über alle Werte von k und die Fenstergröße, für die die Klassifikation am besten funktioniert hat. Außerdem berechnen wir den balancierten relativen Fehler für die vollständige Merkmalsmenge, bevor die Merkmalsauswahl darauf angewendet wird (s. Abschnitt 3.1.2), und den Mittelwert des balancierten relativen Fehlers für jeweils jede Merkmalsgruppe. Die so berechneten Werte stehen in den Tabellen 3.1 und 3.2.

CDs							
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B	Gesamt
Chroma	0,160 (32s)	0,297 (8s)	0,280 (8s)	0,345 (2s)	0,142 (2s)	0,415 (8s)	0,273
Chroma-verwandt	0,101 (4s)	0,338 (2s)	0,391 (4s)	0,358 (2s)	0,298 (2s)	0,325 (2s)	0,302
Harmonie	0,096 (4s)	0,397 (8s)	0,224 (8s)	0,389 (2s)	0,408 (4s)	0,363 (4s)	0,313
Klangfarbe	0,138 (8s)	0,311 (4s)	0,349 (8s)	0,361 (2s)	0,336 (2s)	0,380 (4s)	0,312
Tempo und Rhythmus	0,169 (2s)	0,274 (2s)	0,373 (4s)	0,457 (2s)	0,402 (64s)	0,428 (16s)	0,351
Akkordvektor	0,252 (32s)	0,292 (8s)	0,245 (8s)	0,340 (8s)	0,378 (2s)	0,400 (64s)	0,318
Instrumente	0,092 (16s)	0,468 (32s)	0,252 (16s)	0,414 (64s)	0,356 (64s)	0,451 (16s)	0,339
Akkorde	0,428 (16s)	0,409 (16s)	0,420 (16s)	0,468 (64s)	0,488 (16s)	0,487 (32s)	0,450
Vollst. Merkmalsmenge	0,102	0,289	0,244	0,338	0,139	0,477	0,265

Tabelle 3.1: Balancierter relativer Fehler für alle Kombinationen aus Genres und Merkmalsgruppen mit der jeweils besten Fenstergröße auf der CDs-Datenbank. Die verwendete Fenstergröße steht in Klammern.

1517							
	Classical	Electr. & D.	Jazz	Pop/Rock	Hip-Hop	R'n'B & S.	Gesamt
Chroma	0,238 (8s)	0,262 (2s)	0,362 (8s)	0,409 (32s)	0,190 (4s)	0,365 (8s)	0,304
Chroma-verwandt	0,265 (2s)	0,374 (16s)	0,383 (64s)	0,321 (2s)	0,336 (2s)	0,362 (2s)	0,340
Harmonie	0,192 (8s)	0,306 (2s)	0,433 (8s)	0,445 (2s)	0,271 (4s)	0,391 (4s)	0,339
Klangfarbe	0,278 (8s)	0,320 (4s)	0,420 (8s)	0,348 (2s)	0,371 (8s)	0,419 (2s)	0,359
Tempo und Rhythmus	0,303 (2s)	0,285 (16s)	0,442 (64s)	0,462 (32s)	0,389 (64s)	0,459 (8s)	0,390
Akkordvektor	0,360 (2s)	0,283 (4s)	0,383 (16s)	0,444 (16s)	0,297 (2s)	0,410 (2s)	0,363
Instrumente	0,262 (16s)	0,444 (64s)	0,369 (16s)	0,350 (16s)	0,346 (16s)	0,431 (32s)	0,367
Akkorde	0,428 (32s)	0,386 (64s)	0,450 (32s)	0,460 (64s)	0,405 (16s)	0,471 (32s)	0,433
Vollst. Merkmalsmenge	0,234	0,279	0,331	0,399	0,184	0,360	0,298

Tabelle 3.2: Balancierter relativer Fehler für alle Kombinationen aus Genres und Merkmalsgruppen mit der jeweils besten Fenstergröße auf der 1517-Datenbank. Die verwendete Fenstergröße steht in Klammern.

Außerdem berechnen wir für jede Merkmalsgruppe die durchschnittliche strukturelle Komplexität für die verschiedenen Fenstergrößen und Genres, um besser nachvollziehen zu können, warum eine Merkmalsgruppe für bestimmte Genres gut funktioniert.

Die Merkmalsgruppe Chroma hat auf beiden Datenbanken den deutlich besten Klassifikationsfehler, was sich dadurch erklären lassen kann, dass die Chroma-Komplexitäten der einzelnen Genres zumindest bei einigen Fenstergrößen ziemlich weit auseinander liegen (s. Abbildung 3.1). Chroma ist die mit Abstand beste Merkmalsgruppe zur Klassifikation der Genres Rap (CDs) und Hip-Hop (1517). Chroma scheint sich also gut dazu zu eignen, diese Art von Musik zu klassifizieren. Wie man auf den Abbildungen erkennt, haben diese Genres auf ihrer jeweiligen Datenbank für alle Fenstergrößen die geringste Komplexität. Auch für die Genres Classical (CDs) und Classical (1517), erreicht diese Merkmalsmenge einen guten Klassifikationsfehler, auch wenn andere Merkmale auf diesen Genres besser funktionieren. Wie man an der Abbildung 3.1 sieht, haben diese Genres auf ihrer jeweiligen Datenbank eine deutlich höhere Chroma-Komplexität als die anderen Genres. Allerdings

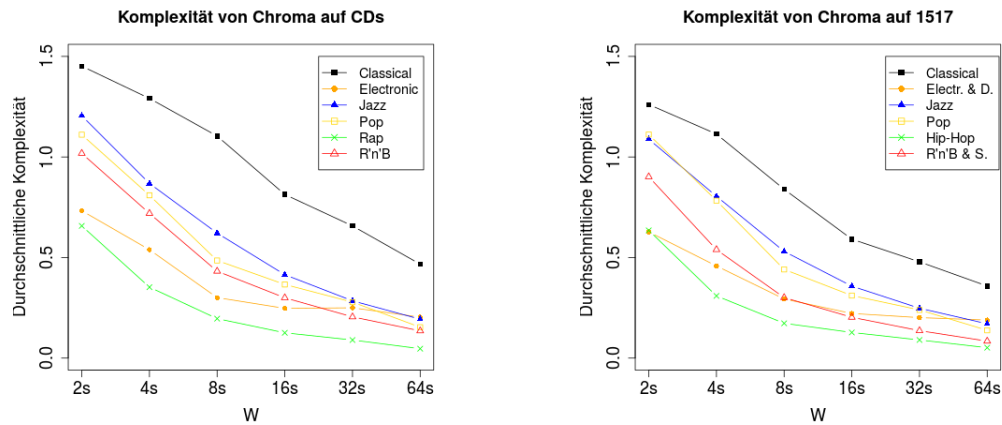


Abbildung 3.1: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Chroma auf der CDs- und der 1517-Datenbank.

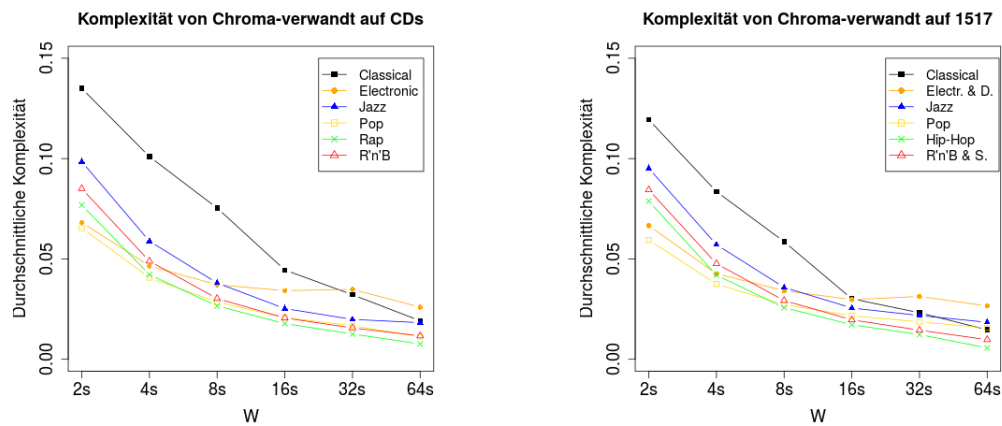


Abbildung 3.2: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Chroma-verwandt auf der CDs- und der 1517-Datenbank.

ist auf der 1517-Datenbank die Klassifikation des Genres Classical deutlich schlechter als die des Genres Hip-Hop. Insgesamt fällt auf, dass auf der 1517-Datenbank klassische Musik deutlich schlechter klassifiziert wurde als auf der CDs-Datenbank. Auf der 1517-Datenbank lassen sich außerdem die Genres Electronic and Dance und Jazz am besten mit Chroma klassifizieren. Dennoch wird auch mit Chroma kein besonders guter Fehler für Jazz erreicht. Dieses Genre scheint insgesamt nur schwer zu klassifizieren zu sein.

Die Merkmalsmenge Chroma-verwandt hat für die Genres Classical (CDs) und Classical (1517) die besten Klassifikationsfehler erreicht. Wie bei den meisten Merkmalsmengen hat klassische Musik hier für die meisten Fenstergrößen eine deutlich höhere strukturelle Komplexität als die anderen Genres (s. Abbildung 3.2). Auf beiden Datenbanken lässt sie sich allerdings mit anderen Merkmalen besser klassifizieren. Auf der CDs Datenbank lässt sich das Genre R'n'B mit der Merkmalsgruppe Chroma-verwandt am besten, trotzdem

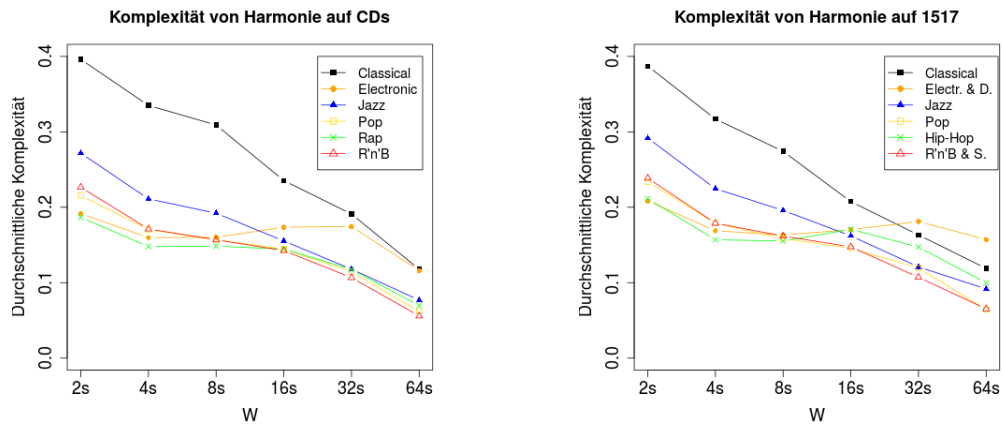


Abbildung 3.3: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Harmonie auf der CDs- und der 1517-Datenbank.

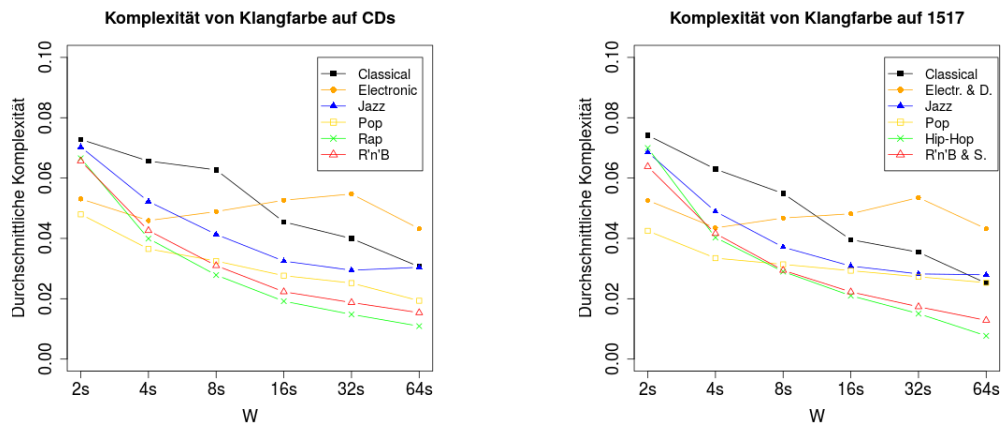


Abbildung 3.4: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Klangfarbe auf der CDs- und der 1517-Datenbank.

aber nicht sehr gut, klassifizieren. Auf der 1517-Datenbank lassen sich die Genres R'n'B and Soul und Pop/Rock am besten mit dieser Merkmalsgruppe klassifizieren. Beide lassen sich aber auch so nicht besonders gut klassifizieren und schneiden insgesamt eher schlecht ab.

Wie die meisten Merkmalsgruppen erreicht die Merkmalsgruppe Harmonie die besten Ergebnisse für die Genres Classical (CDs) und Classical (1517), die für die meisten Fenstergößen die größte Komplexität erreichen (s. Abbildung 3.3). Die Merkmalsgruppe erreicht aber auch im Vergleich zu anderen Merkmalsgruppen sehr gute Ergebnisse für diese Genres. Auf der 1517-Datenbank ist sie sogar die Merkmalsgruppe, die das Genre Classical am besten klassifiziert. Auf der CDs-Datenbank ist sie die beste Merkmalsmenge für das Genre Jazz, schneidet aber auf der 1517-Datenbank recht schlecht für Jazz ab.

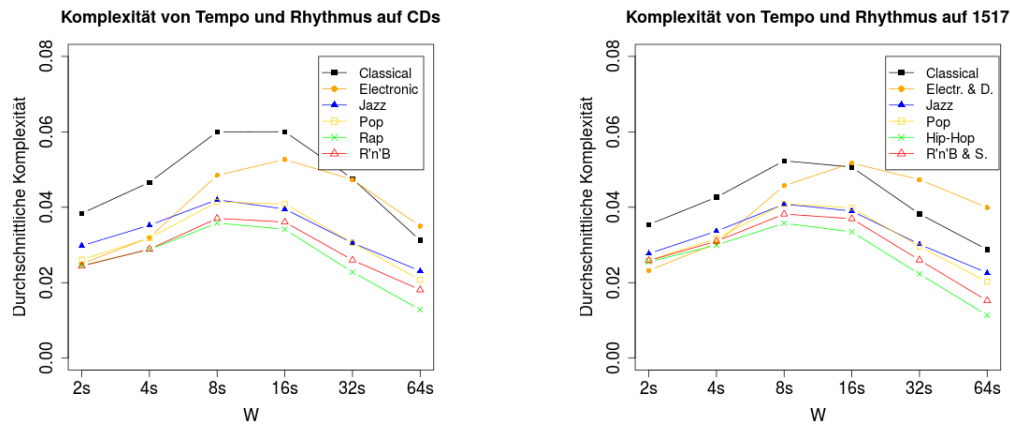


Abbildung 3.5: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Tempo und Rhythmus auf der CDs- und der 1517-Datenbank.

Auch die Merkmalsgruppe Klangfarbe hat auf beiden Datenbanken für klassische Musik die besten Ergebnisse erzielt. Die zweitbesten Ergebnisse erreichte sie auf beiden Datenbanken für elektronische Musik, also für die Genres Electronic (CDs) und Electronic and Dance (1517). Man könnte vermuten, dass sich die vergleichsweise guten Ergebnisse für elektronische Musik durch deren hohe Komplexität bei großen Fenstergrößen erklären lässt (s. Abbildung 3.4). Interessanterweise wird aber auf beiden Datenbanken das beste Ergebnis für Zeitfenster der Größe 4s erreicht. Bei der Interpretation der Abbildungen, die die strukturelle Komplexität der verschiedenen Genres darstellen, muss man bedenken, dass sie nur die durchschnittliche Komplexität darstellen. Dem Klassifikator stehen aber auch Informationen, wie Minimum, Maximum, Varianz und andere zur Verfügung (s. Abschnitt 2.2.5). Insgesamt erreicht diese Merkmalsgruppe allerdings keine sehr guten Ergebnisse.

Die Merkmalsgruppe Tempo und Rhythmus erreicht auf der CDs-Datenbank, wie die meisten Merkmalsgruppen, ihre besten Ergebnisse für das Genre Classical. Außerdem ist sie die Merkmalsgruppe, die für das Genre Electronic die besten Ergebnisse erzielt. Wie man auf Abbildung 3.5 sieht, hat dieses Genre für einige Fenstergrößen eine recht hohe Komplexität. Besonders deutlich scheint es sich für mittelgroße Fenstergrößen abzuheben, dennoch wird der geringste Fehler für die Fenstergröße 2s erreicht. Auf der 1517-Datenbank erreicht die Merkmalsgruppe ihre besten Ergebnisse für das Genre Electronic and Dance. Allerdings erreichen andere Merkmalsgruppen hier noch geringere Fehler. Auch auf der 1517-Datenbank hat die elektronische Musik für einige Fenstergrößen eine recht hohe Komplexität von Tempo und Rhythmus. Bei großen Fenstergrößen sogar deutlich höher als Classical. Bei kleinen Fenstergrößen hat es wiederum eine sehr geringe Komplexität.

Das Merkmal Akkordvektor erreicht auf der CDs-Datenbank sein bestes Ergebnis bei dem Genre Jazz. Wie man an Abbildung 3.7 erkennt, hat dieses Genre eine recht große

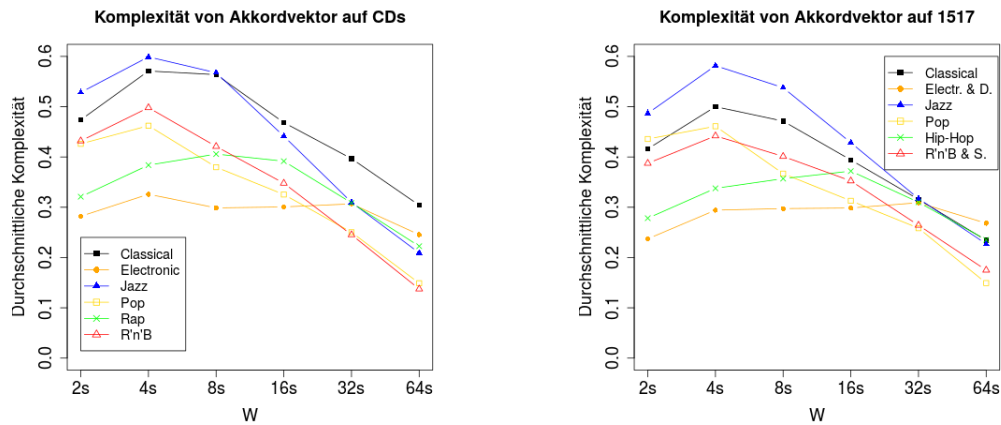


Abbildung 3.6: Durchschnittliche strukturelle Komplexität des Merkmals Akkordvektor auf der CDs- und der 1517-Datenbank.

Komplexität für kleine Fenstergrößen. Das beste Ergebnis wird bei einer Fenstergröße von 8s erreicht. Auch für Classical und Electronic erreicht es relativ gute Ergebnisse. Wie man an der Abbildung erkennen kann, hat Classical für große Fenstergrößen eine höhere Komplexität als die anderen Genres. Electronic hat für kleine Fenstergrößen eine geringere Komplexität als die anderen Genres. So werden für Classical die besten Ergebnisse bei einer Fenstergröße von 32s und für Electronic mit einer Fenstergröße von 8s erreicht. Für Classical schneidet Akkordvektor aber deutlich schlechter als die meisten anderen Merkmalsgruppen ab. Es hat einen recht großen Fehler beim Genre Pop/Rock, trotzdem ist es das Merkmal, das am besten auf diesem Genre abschneidet, welches für alle Merkmale große Fehler hat. Auf der 1517-Datenbank erzielt das Akkordvektor-Merkmal die besten Ergebnisse für die Genres Electronic and Dance und Hip-Hop, die für kleine Fenstergrößen eine deutlich kleinere Komplexität als die anderen Genres haben. So sind die besten Fenstergrößen zur Klassifikation 4s für Electronic and Dance und 2s für Hip-Hop. Auf dem Genre Hip-Hop ist es allerdings deutlich schlechter als die Merkmalsgruppe Chroma. Auch auf dieser Datenbank hat das Genre Jazz eine hohe Komplexität. Dennoch lässt es sich nicht sehr gut klassifizieren. Auf der 1517-Datenbank lässt sich Jazz allerdings insgesamt nur schlecht klassifizieren. Im Vergleich zu den anderen Merkmalen schneidet Akkordvektor noch recht gut ab.

Die Merkmalsgruppe Instrumente erreicht auf der CDs-Datenbank ihr bestes Ergebnis für das Genre Classical. Sie ist die Merkmalsgruppe, die auf dieser Merkmalsgruppe, den geringsten Fehler erreicht. Wie man an der Abbildung 3.7 sieht, hat dieses Genre, eine deutlich höhere Komplexität als die anderen Genres. Dies bedeutet, dass sich die Anteile der vier Instrumentengruppen, die diese Merkmalsgruppe erkennen kann (s. Abschnitt 2.1), in klassischer Musik besonders stark verändern. Auch für Jazz erreicht sie ein recht gutes Ergebnis. Jazz hat zwar eine deutlich niedrige Komplexität als Classic, aber

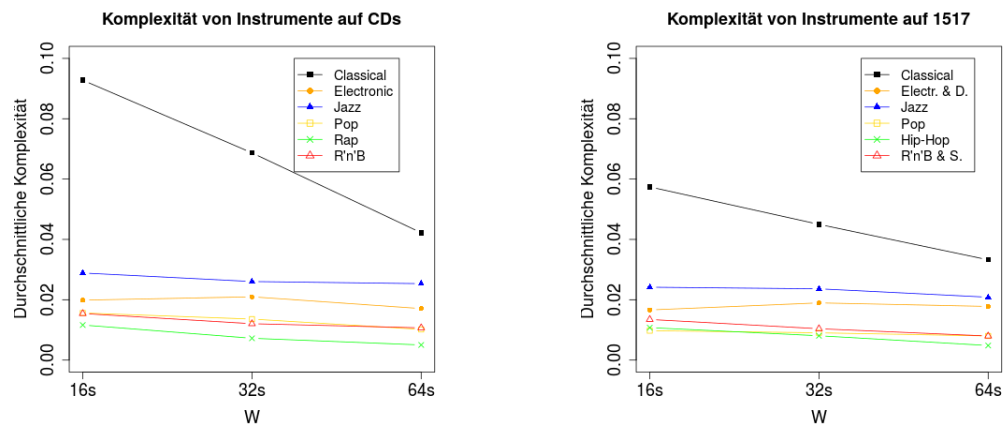


Abbildung 3.7: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Instrumente auf der CDs- und der 1517-Datenbank.

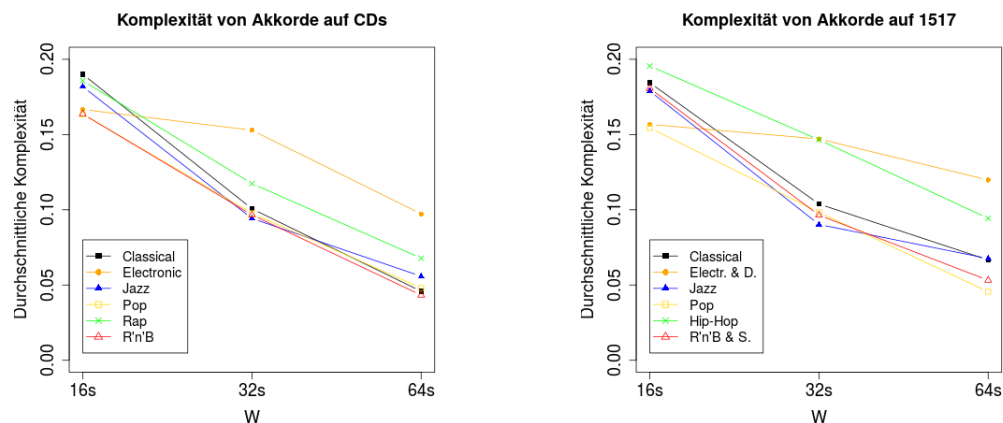


Abbildung 3.8: Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Akkorde auf der CDs- und der 1517-Datenbank.

dennoch eine höhere als die anderen Genres. Auch auf der 1517-Datenbank erreicht diese Merkmalsgruppe ihr bestes Ergebnis für das Genre Classical. Es gibt allerdings einige andere Merkmalsgruppen, die bessere Ergebnisse auf diesem Genre erzielen.

Die Merkmalsgruppe Akkorde erreicht ihre besten Ergebnisse für die Genres Electronic (CDs) und Electronic and Dance (1517), die eine recht hohe Komplexität im Vergleich zu den anderen Genres haben (s. Abbildung 3.8). Insgesamt erreicht diese Merkmalsgruppe allerdings keine guten Ergebnisse, auch nicht für elektronische Musik. Sie ist die Merkmalsgruppe, die auf beiden Datenbanken durchschnittlich die schlechtesten Ergebnisse erzielt. Wie man in Abbildung 3.8 sieht, ist die Komplexität der unterschiedlichen Genres auch sehr ähnlich.

Insgesamt scheint die Klassifikation mit der vollständigen Merkmalsmenge zu den besten Ergebnissen zu führen. Dies ist auch insofern verständlich, da nur dann Zusammenhänge zwischen verschiedenen Merkmalsgruppen und Fenstergrößen ausgenutzt werden können. Bei allen anderen bis jetzt untersuchten Einstellungen wurden schließlich nur jeweils eine Merkmalsgruppe und eine Fenstergröße genutzt. Trotzdem führte auch das in einigen Situationen bereits zu guten Ergebnissen; besonders bei den Genres Classical und Rap (CDs) bzw. Classical und Hip-Hop (1517). Insgesamt fällt auf, dass die Leistung der unterschiedlichen Merkmalsgruppen stark von dem zu untersuchenden Genre abhängen. Außerdem ist es interessant zu sehen, dass das Merkmal Akkordvektor, welches wir uns für diese Arbeit als alternative zur Merkmalsgruppe Akkorde überlegt haben, für alle Genres besser als diese Merkmalsgruppe abschneidet. Außerdem fällt auf, dass die Klassifikation auf der CDs-Datenbank für die meisten Genres besser gelingt als auf der 1517-Datenbank. Das könnte daran liegen, dass die 1517-Datenbank viele verschiedene Genres enthält, während in der CDs-Datenbank nur die sechs Genres vorkommen, die wir in dieser Arbeit klassifizieren.

Fenstergrößen

Als nächstes untersuchen wir, welche Fenstergrößen bei der strukturellen Komplexität besonders gut funktionieren. Wie wir bereits im letzten Abschnitt sehen konnte, scheint es stark von der verwendeten Merkmalsgruppe und dem zu klassifizierenden Genre abzuhängen, welche Fenstergröße am besten funktioniert. Das kann man sehr gut sehen, wenn man in den Abbildungen 3.1 und 3.2 schaut, welche Fenstergrößen bei den unterschiedlichen Kombinationen aus Merkmalsgruppe und Genre am besten funktioniert haben. Um die unterschiedlichen Fenstergrößen noch genauer zu analysieren, haben wir in den Tabellen 3.3 und 3.4 für alle Kombinationen aus Genre und Fenstergröße den Mittelwert des balancierten relativen Fehlers auf den beiden Datenbanken berechnet.

CDs						
	2s	4s	8s	16s	32s	64s
Chroma	0,462	0,433	0,415	0,468	0,477	0,497
Chroma-verwandt	0,325	0,366	0,415	0,389	0,483	0,487
Harmonie	0,379	0,363	0,495	0,474	0,395	0,496
Klangfarbe	0,393	0,380	0,482	0,475	0,391	0,486
Tempo und Rhythmus	0,498	0,483	0,518	0,428	0,447	0,443
Akkordvektor	0,490	0,461	0,481	0,514	0,469	0,400
Instrumente	-	-	-	0,451	0,453	0,492
Akkorde	-	-	-	0,494	0,487	0,523

Tabelle 3.3: Mittelwerte des balancierten relativen Fehlers für alle Kombinationen aus Genre und Fenstergröße auf der CDs-Datenbank.

1517						
	2s	4s	8s	16s	32s	64s
Chroma	0,409	0,404	0,365	0,442	0,442	0,422
Chroma-verwandt	0,362	0,419	0,426	0,479	0,434	0,516
Harmonie	0,404	0,391	0,437	0,490	0,451	0,459
Klangfarbe	0,419	0,423	0,439	0,436	0,457	0,457
Tempo und Rhythmus	0,486	0,487	0,459	0,470	0,469	0,470
Akkordvektor	0,410	0,484	0,530	0,498	0,488	0,532
Instrumente	-	-	-	0,436	0,431	0,496
Akkorde	-	-	-	0,487	0,471	0,475

Tabelle 3.4: Mittelwerte des balancierten relativen Fehlers für alle Kombinationen aus Genre und Fenstergröße auf der 1517-Datenbank.

Mit der Merkmalsgruppe Chroma werden auf beiden Datenbanken die besten durchschnittlichen Ergebnisse für die Fenstergröße 8s erreicht (s. Tabellen 3.3 und 3.4). Für einzelne Genres werden aber zum Teil bessere Ergebnisse für andere Fenstergrößen erzielt (s. Tabellen 3.1 und 3.2).

Bei der Merkmalsgruppe Chroma-verwandt werden die deutlich besten Ergebnisse für die Fenstergröße 2s erzielt und auch 4s hat durchschnittlich deutlich bessere Ergebnisse als die anderen Fenstergrößen. Bei allen Genres wurde der geringste Klassifikationsfehler entweder für 2s oder 4s erreicht außer für die Genres Electronic and Dance und Jazz auf der 1517-Datenbank (s. Tabellen 3.1 und 3.2). Wie man an Abbildung 3.9 sehen kann, unterscheiden sich bei diesen Genres die Werte kaum für die verschiedenen Fenstergrößen. Für die anderen Genres scheint die Klassifikation für kleine Zeitfenster deutlich besser zu

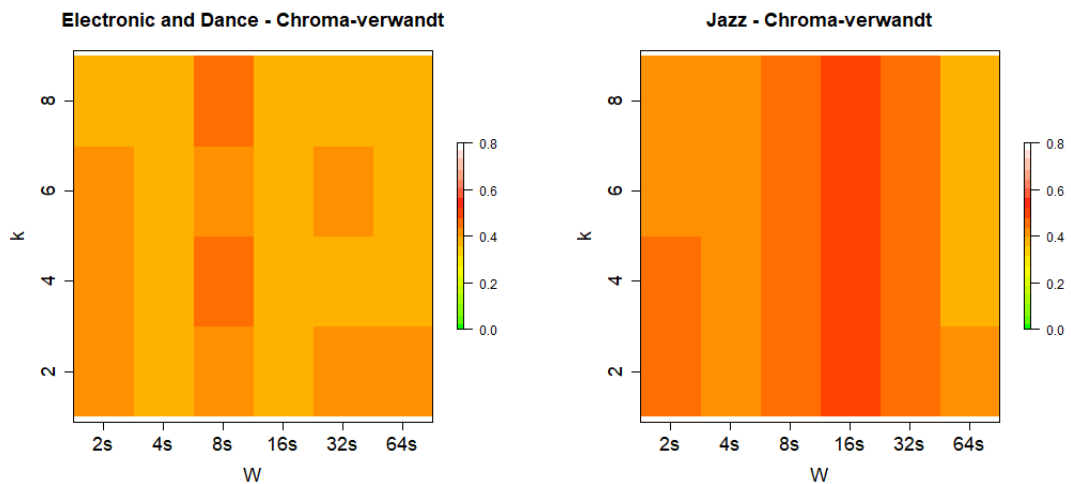


Abbildung 3.9: m_{BRE} für die Genres Electronic and Dance und Jazz der 1517-Datenbank für alle Werte von k und W und die Merkmalsgruppe Chroma-verwandt.

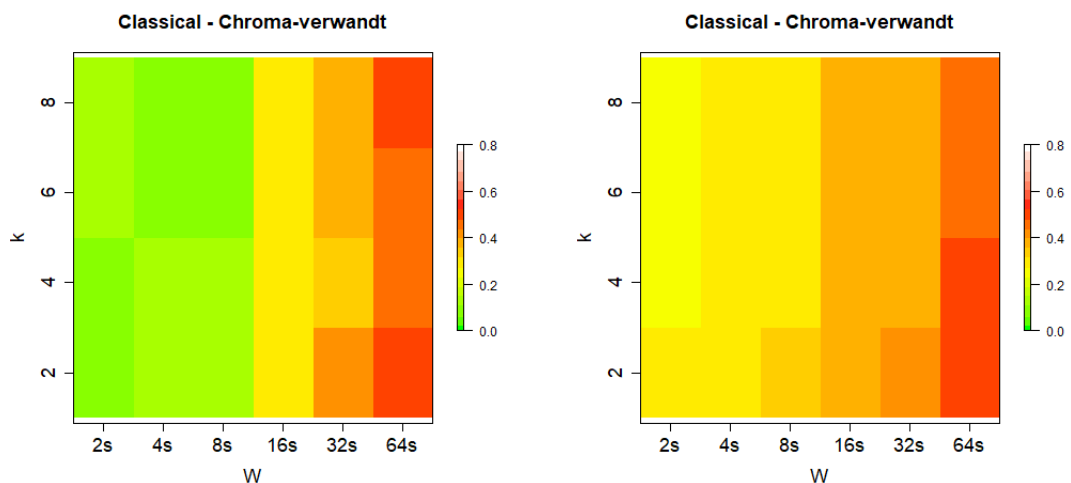


Abbildung 3.10: m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Chroma-verwandt.

funktionieren als für große. Besonders deutlich sieht man dies bei der klassischen Musik (s. Abbildung 3.10).

Für die Merkmalsgruppe Harmonie werden auf der CDs-Datenbank die deutlich besten Durchschnittsergebnisse für die Fenstergrößen 2s, 4s und 32s erzielt. Wenn man sich die Ergebnisse auf den beiden Genres Classical und Jazz ansieht, auf denen diese Merkmalsgruppe zu vergleichsweise guten Ergebnissen geführt hat, sieht man, dass dort kleine Fenstergrößen zu den besten Ergebnissen führen (s. Abbildung 3.11). Interessanterweise werden bei Classical die besten Ergebnisse für die untere Hälfte der Fenstergrößen erzielt, während sich bei Jazz die Fenstergröße 8s deutlich von den anderen als beste Fenstergrö-

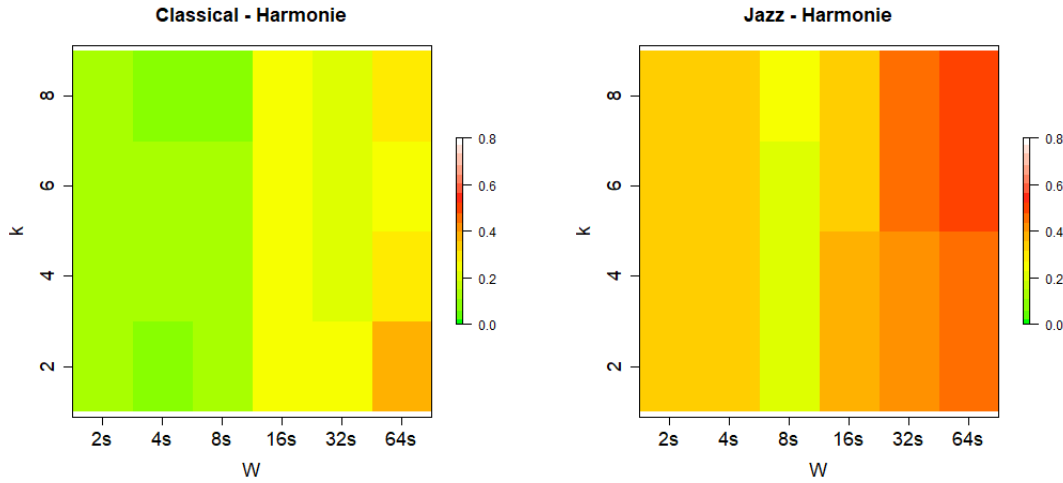


Abbildung 3.11: m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Harmonie.

ße abhebt. Für alle Genres der CDs-Datenbank werden die besten Ergebnisse für eine Fenstergröße von höchstens 8s erzielt (s. Tabelle 3.1). Bei der 1517-Datenbank werden durchschnittlich die besten Ergebnisse für die kleinen Fenstergrößen 2s, 4s und 8s erzielt. Auch für die einzelnen Genres dieser Datenbank wird das beste Ergebnis für jeweils eine dieser kleineren Fenstergrößen erzielt (s. Tabelle 3.2).

Die Ergebnisse für die Merkmalsgruppe Klangfarbe sind ähnlich. Die besten Durchschnittsergebnisse werden für die CDs-Datenbank bei den Fenstergrößen 2s, 4s und 32s erzielt und für die 1517-Datenbank bei den Fenstergrößen 2s und 4s. Für die einzelnen Genres beider Datenbanken werden die jeweils besten Ergebnisse für 2s, 4s oder 8s erzielt (s. Tabellen 3.1 und 3.2). Bei dem Genre Classical der CDs-Datenbank, auf dem diese Merkmalsmenge die besten Ergebnisse erreicht, funktioniert die Fenstergröße 8s deutlich am besten. Ähnlich ist es bei dem analogen Genre Classical der 1517-Datenbank (s. Abbildung 3.12).

Bei der Merkmalsgruppe Tempo und Rhythmus wird das durchschnittlich beste Ergebnis auf der CDs-Datenbank für $W = 16s$ und auf der 1517-Datenbank für $W = 32s$ erreicht. Insgesamt hat diese Merkmalsgruppe aber eher schlechte Ergebnisse. Nur bei elektronischer Musik hat sie recht gute Ergebnisse erreicht. Auf der CDs-Datenbank wurden dort die besten Ergebnisse für die Fenstergröße 2s und auf der 1517-Datenbank für 16s erreicht. Insgesamt scheint es aber besser für kleinere Fenstergrößen als für größere zu funktionieren (s. Abbildung 3.13).

Die Merkmalsgruppe Akkordvektor erreicht auf der CDs-Datenbank die durchschnittlich besten Ergebnisse für die Fenstergröße 64s und auf der CDs-Datenbank und für 2s auf der 1517-Datenbank. Insgesamt scheint es stark vom Genre abzuhängen, welche Fenster-

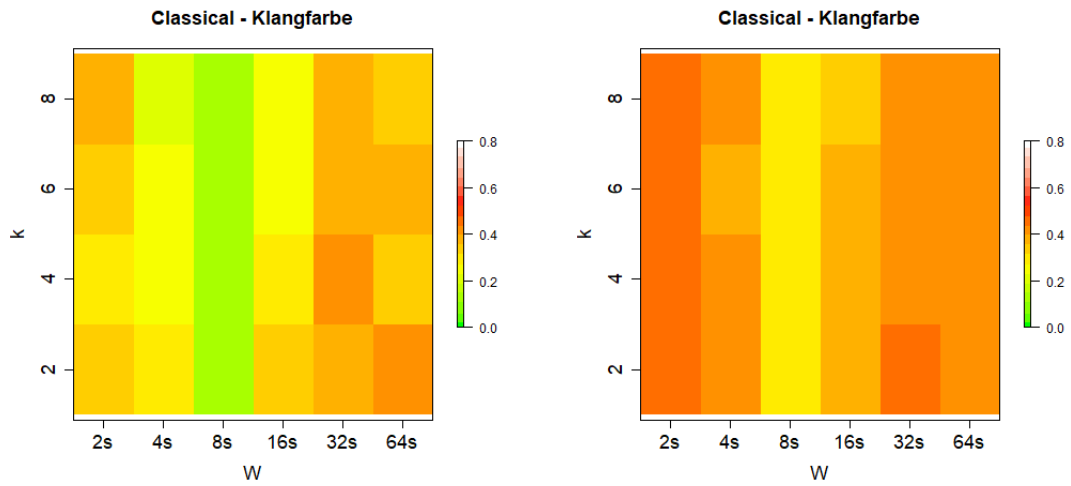


Abbildung 3.12: m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Klangfarbe.

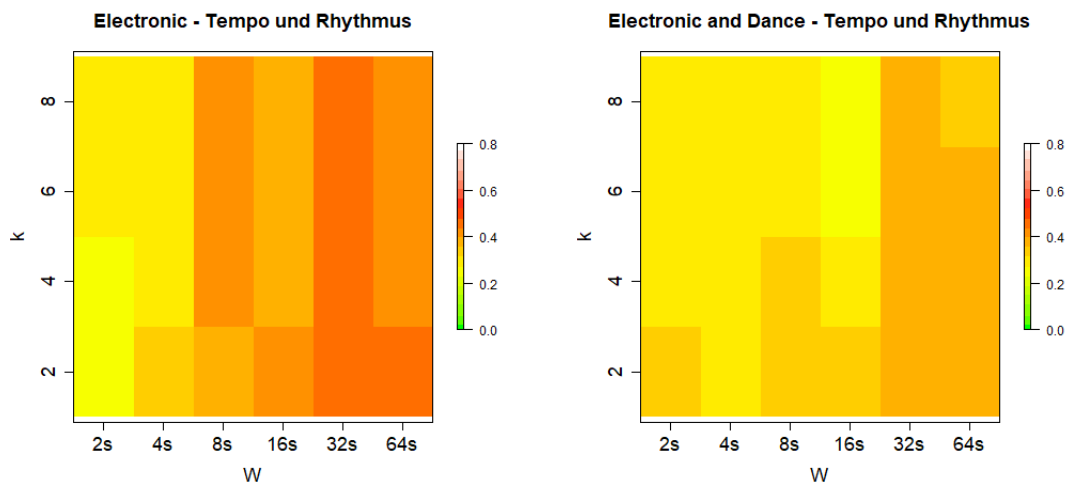


Abbildung 3.13: m_{BRE} für die Genres Electronic (CDs) und Electronic and Dance (1517) für alle Werte von k und W und die Merkmalsgruppe Tempo und Rhythmus.

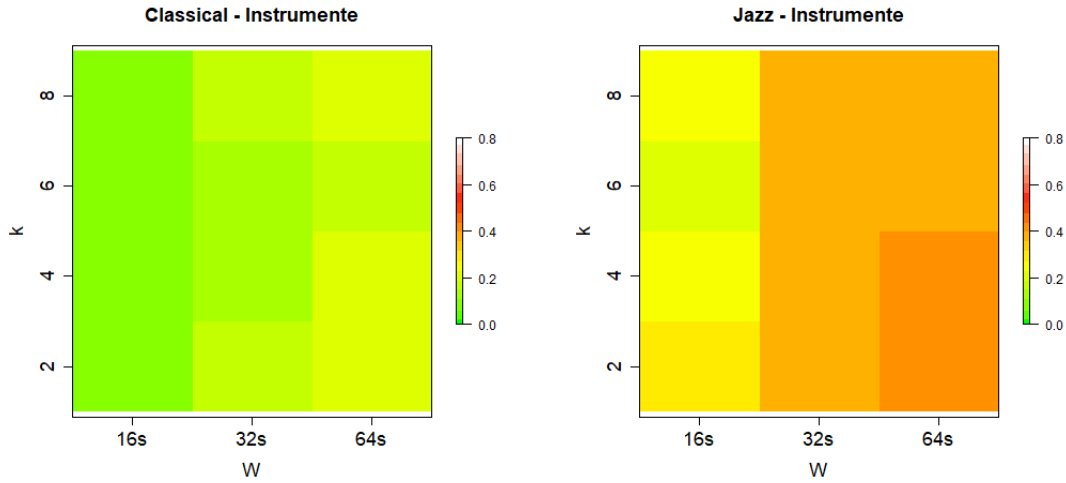


Abbildung 3.14: m_{BRE} für die Genres Classical und Jazz der CDs-Datenbank für alle Werte von k und W und die Merkmalsgruppe Tempo und Rhythmus.

größe am besten funktioniert, es fällt aber auf, dass auf der 1517-Datenbank eher kleinere Fenstergrößen als auf der CDs-Datenbank gut funktionieren (s. Tabellen 3.1 und 3.2).

Die Merkmalsgruppe Instrumente erreicht auf der CDs-Datenbank für kürzere Fenstergrößen durchschnittlich bessere Ergebnisse. Besonders deutlich wird dies für die beiden Genres, bei denen sie die besten Ergebnisse erzielt, Classical und Jazz (s. Abbildung 3.14). Auf der 1517-Datenbank werden die durchschnittlich besten Ergebnisse für die Fenstergröße 32s erzielt. Für die meisten Genre werden aber auch hier die besten Ergebnisse für die kleinste Fenstergröße 16s erzielt, besonders für die Genres, auf denen diese Merkmalsgruppe gute Ergebnisse erzielt, wie Classical und Pop/Rock (s. Tabelle 3.2).

Die Merkmalsgruppe Akkorde erzielt bei beiden Datenbanken durchschnittlich die besten Ergebnisse für $W = 16s$. Mit dieser Merkmalsgruppe konnten allerdings für alle Genres und Fenstergrößen keine besonders guten Ergebnisse erzielt werden (s. Tabellen 3.1 und 3.2).

Es hängt zwar stark davon ab, welche Merkmalsmenge verwendet wird, und welches Genre erkannt werden soll, um zu entscheiden, welche Fenstergröße sich am besten eignet, insgesamt kann man aber sagen, dass die kleinen Fenstergrößen meistens zu besseren Ergebnissen führen als die großen, besonders dann, wenn man für jedes Genre die Merkmalsgruppen betrachtet, die für das Genre zu guten Ergebnissen führen. Das lässt sich dadurch erklären, dass bei großen Zeitfenstern die Merkmale über sehr große Zeiträume zusammengefasst werden und die Stücke in große Partitionen eingeteilt werden. In [20] zeigte sich, dass das der Klassifikationsqualität schadet.

3.2.2 FKNN

Im folgenden Abschnitt untersuchen wir die Ergebnisse unserer Experimente im Hinblick auf den FKNN-Algorithmus, den wir zur Klassifikation verwendet haben (s. Abschnitt 2.2.4). Wir untersuchen dabei den Parameter k , der die Anzahl der nächsten Nachbarn bestimmt und die Zugehörigkeitsgrade, die FKNN als Ausgabe zurückgibt.

Werte für den Parameter k

Um herauszufinden, welchen Einfluss der Wert von k auf die Klassifikation hat, berechnen wir für jeden getesteten Wert von k den durchschnittlichen balancierten relativen Fehler. Zuerst berechnen wir für jeden Wert von k den Mittelwert von m_{BRE} für alle Klassifikationen, die wir durchgeführt haben, also den Mittelwert über die acht verwendeten Merkmalsmengen, alle Genres und alle Fenstergrößen. Danach berechnen wir für jedes k den Mittelwert über alle Merkmalsmengen und Genres jeweils für die Fenstergröße, die den geringsten Klassifikationsfehler hat. Das tun wir deshalb, weil die Größe der Zeitfenster eine große Auswirkung auf die Klassifikation hat, wie wir in Abschnitt 3.2.1 gesehen haben.

CDs		
	Alle Fenstergrößen	Beste Fenstergröße
$k = 2$	0,397	0,335
$k = 4$	0,390	0,328
$k = 6$	0,386	0,326
$k = 8$	0,385	0,327

Tabelle 3.5: Mittelwerte des balancierten relativen Fehlers für verschiedene Werte für k auf der CDs-Datenbank

1517		
	Alle Fenstergrößen	Beste Fenstergröße
$k = 2$	0,416	0,368
$k = 4$	0,409	0,360
$k = 6$	0,405	0,358
$k = 8$	0,404	0,356

Tabelle 3.6: Mittelwerte des balancierten relativen Fehlers für verschiedene Werte für k auf der 1517-Datenbank

Auf der CDs-Datenbank, werden die besten Klassifikationsleistungen bei $k = 8$ für alle Fenstergrößen zusammen und bei $k = 6$ bei den besten Fenstergrößen erzielt. Auf der 1517-Datenbank werden die besten Ergebnisse bei $k = 8$ erzielt. Bei der CDs-Datenbank ist allerdings kein großer Unterschied zwischen $k = 6$ und $k = 8$ zu sehen. Insgesamt scheint die Klassifikation für größere k etwas besser zu funktionieren. Eine große Auswirkung scheint der Wert von k allerdings nicht zu haben. Dies lässt sich auch schon recht gut an den Abbildungen im letzten Abschnitt erkennen. Das lässt sich so erklären, dass größere k -Werte dafür sorgen, dass mehr Informationen von Trainingsbeispielen verwendet werden. Dadurch kann man bei einem etwas größeren Wert für k eine genauere Klassifikation erzielen. Viel wird das Ergebnis vermutlich deshalb nicht beeinflusst, da der FKNN-Algorithmus, die weiter entfernten Beispiele, die durch größere k -Werte hinzukommen weniger stark gewichtet (s. Abschnitt 2.3.2).

Zugehörigkeitsgrade

Nachdem wir den Parameter k des FKNN-Algorithmus angeschaut haben, betrachten wir nun die Zugehörigkeitsgrade, die der Algorithmus als Ausgabe der Klassifikation liefert. Keller, Gray und Givens haben bei ihren Untersuchungen dieses Algorithmus die Beobachtung gemacht, dass sich die Zugehörigkeitsgrade gut als Sicherheit der Klassifikation deuten lassen. So erhielten falsch klassifizierte Instanzen relativ niedrige Zugehörigkeitsgrade für die verschiedenen Klassen [8]. Um zu untersuchen, ob FKNN Zugehörigkeitsgrade liefert, die die Sicherheit der Klassifikationen sinnvoll widerspiegeln, haben wir für die verschiedenen Genres die Mittelwerte der Zugehörigkeitsgrade berechnet, mit denen der Algorithmus, wahre Positive (TP), falsche Positive (FP), wahre Negative (TN) und falsche Negative (FN) (s. Abschnitt 2.4.2) klassifiziert. Wir haben dazu den Algorithmus mit der vollständigen Merkmalsmenge und $k = 8$ angewendet. Die Mittelwerte findet man in den Tabellen 3.7 und 3.8.

CDs						
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B
TP	0,828	0,657	0,701	0,814	0,858	0,654
FP	0,700	0,641	0,600	0,704	0,738	0,618
TN	0,113	0,209	0,284	0,337	0,130	0,003
FN	0,380	0,297	0,412	0,426	0,224	0,323

Tabelle 3.7: Durchschnittlich durch FKNN berechnete Zugehörigkeitsgrade für wahre Positive (TP), falsche Positive (FP), wahre Negative (TN) und falsche Negative (FN) bei Klassifikationen mit der vollständigen Merkmalsmenge und $k = 8$ auf der CDs-Datenbank

	1517					
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B
<i>TP</i>	0,820	0,746	0,730	0,672	0,817	0,723
<i>FP</i>	0,680	0,655	0,648	0,645	0,732	0,673
<i>TN</i>	0,212	0,236	0,316	0,320	0,142	0,295
<i>FN</i>	0,342	0,306	0,331	0,383	0,293	0,366

Tabelle 3.8: Durchschnittlich durch FKNN berechnete Zugehörigkeitsgrade für wahre Positive (*TP*), falsche Positive (*FP*), wahre Negative (*TN*) und falsche Negative (*FN*) bei Klassifikationen mit der vollständigen Merkmalsmenge und $k = 8$ auf der 1517-Datenbank

Wie man an den Tabellen sieht, erhalten die fälschlicherweise als positiv klassifizierten Stücke für alle Genres geringere Zugehörigkeitsgrade als die wahren Positiven. Genauso erhalten die falschen Negativen für alle Genres größere Zugehörigkeitsgrade als die wahren Negativen. Man kann dies also so deuten, dass der Algorithmus sich bei den falschen Klassifikationen im Schnitt weniger sicher ist. Die Zugehörigkeitsgrade scheinen also tatsächlich eine Art von Sicherheit der Klassifikation widerzuspiegeln.

Da wir jedes Genre einzeln binär klassifizieren, ist es möglich, dass ein Stück als zu mehreren Genres zugehörig erkannt wird, obwohl die Stücke, die wir verwenden, tatsächlich zu jeweils genau einem Genre gehören. In den Tabellen 3.9 und 3.10 kann man sehen, welcher Anteil der Stücke der verschiedenen Genres zu den untersuchten Genres klassifiziert wurde. Wir haben zur Klassifikation wieder die vollständige Merkmalsmenge mit $k = 8$ verwendet. Die tatsächlichen Genres werden durch die Reihen und die zu klassifizierenden Genres durch die Spalten dargestellt. Für die 1517-Datenbank gibt es außerdem die Reihe *Sonstiges*, da nicht alle Stücke einem der sechs untersuchten Genres angehören. Da ein Stück zu mehreren Genres oder auch zu keinem Genre klassifiziert werden kann, müssen die Summen der Spalten nicht 1 sein, sondern sind für gewöhnlich größer. So werden zum Beispiel viele klassische Stücke auch für Jazz als positiv und viele Jazzstücke auch für Pop/Rock als positiv klassifiziert.

Eine Idee, die Zugehörigkeitsgrade zu nutzen, die der FKNN-Algorithmus berechnet, um dieses Problem zu lösen und die Klassifikation zu verbessern, ist, jedes Stück nur zu dem Genre zu klassifizieren zu dem es den höchsten Zugehörigkeitsgrad hat, wenn dieser größer als 0,5 ist, und es ansonsten als zu keinem der untersuchten Genres zugehörig zu klassifizieren (im Folgenden als *Sonstiges* bezeichnet). Das Stück „Mendelssohn Symphony No. 4 Italian-IV, Saltarello Presto“ des Genres Classical aus der CDs-Datenbank erhielt zum Beispiel einen Zugehörigkeitsgrad von 0,858 zu Classical und einen Zugehörigkeitsgrad von 0,512 zu Jazz. Es wird also korrekterweise als zum Genre Classical zugehörig erkannt, allerdings fälschlicherweise auch zum Genre Jazz klassifiziert. Mit unserer Verbes-

CDs (normal)						
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B
Classical	0,808	0,115	0,462	0,385	0,000	0,154
Electronic	0,000	0,600	0,067	0,633	0,467	0,333
Jazz	0,200	0,067	0,700	0,833	0,167	0,400
Pop/Rock	0,090	0,169	0,135	0,876	0,157	0,124
Rap	0,000	0,433	0,067	0,233	0,967	0,700
R'n'B	0,033	0,167	0,267	0,733	0,433	0,333

Tabelle 3.9: Genrekonfusionsmatrix der Klassifikation mit der vollständigen Merkmalsmenge und $k = 8$ auf der CDs-Datenbank. Die Reihen stehen dabei für die tatsächlichen Genres der Stücke und die Spalten für die klassifizierten Genres.

1517 (normal)						
	Classical	Electr. & D.	Jazz	Pop/Rock	Hip-Hop	R'n'B & S.
Classical	0,675	0,175	0,700	0,325	0,075	0,175
Electr. & D.	0,189	0,700	0,489	0,289	0,567	0,544
Jazz	0,322	0,189	0,744	0,478	0,178	0,522
Pop/Rock	0,220	0,100	0,430	0,690	0,130	0,450
Hip-Hop	0,038	0,612	0,388	0,312	0,875	0,888
R'n'B & S.	0,100	0,280	0,530	0,380	0,410	0,720
Sonstiges	0,265	0,223	0,399	0,485	0,195	0,432

Tabelle 3.10: Genrekonfusionsmatrix der Klassifikation mit der vollständigen Merkmalsmenge und $k = 8$ auf der 1517-Datenbank. Die Reihen stehen dabei für die tatsächlichen Genres der Stücke und die Spalten für die klassifizierten Genres.

serungsidee wird es korrekterweise nur zum Genre Classical klassifiziert. In den Tabellen 3.11 und 3.12 kann man den Effekt dieser neuen Klassifikationsmethode sehen. Die Summen der Reihen ergeben nun 1, da jedes Stück zu genau einem Genre klassifiziert wird. In beiden Datenbanken werden deutlich weniger Stücke des Genres Classical als Jazz klassifiziert. Auf der 1517-Datenbank werden auch deutlich weniger Jazz-Stücke fälschlicherweise als Pop/Rock erkannt. Auf der CDs-Datenbank werden nun allerdings mehr Jazz-Stücke als Pop/Rock klassifiziert als zum tatsächlichen Genre Jazz. Besonders schlecht sieht der Effekt bei dem Genre R'n'B der CDs-Datenbank aus, dessen Stücke nun fast nur noch als Pop/Rock oder Rap erkannt werden. Insgesamt werden nun natürlich für alle Genres deutlich weniger Stücke als positiv klassifiziert.

CDs (nur ein Genre pro Stück)							
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B	Sonstiges
Classical	0,731	0,000	0,038	0,192	0,000	0,000	0,038
Electronic	0,000	0,167	0,033	0,433	0,333	0,033	0,000
Jazz	0,100	0,033	0,167	0,400	0,167	0,133	0,000
Pop/Rock	0,045	0,011	0,000	0,809	0,112	0,000	0,022
Rap	0,000	0,067	0,000	0,100	0,800	0,033	0,000
R'n'B	0,033	0,033	0,033	0,633	0,267	0,000	0,000

Tabelle 3.11: Genrekonfusionsmatrix der Klassifikation mit der vollständigen Merkmalsmenge und $k = 8$ auf der CDs-Datenbank. Die Reihen stehen dabei für die tatsächlichen Genres der Stücke und die Spalten für die klassifizierten Genres. Jedes Stück wird dabei zu dem Genre klassifiziert, zu dem es den höchsten Zugehörigkeitsgrad hat, wenn dieser mindestens 0,5 ist.

1517 (nur ein Genre pro Stück)							
	Classical	Electr. & D.	Jazz	Pop/Rock	Hip-Hop	R'n'B & S.	Sonstiges
Classical	0,500	0,025	0,250	0,050	0,050	0,000	0,125
Electr. & D.	0,111	0,367	0,078	0,089	0,178	0,167	0,011
Jazz	0,144	0,044	0,411	0,100	0,078	0,178	0,044
Pop/Rock	0,080	0,050	0,100	0,410	0,060	0,210	0,090
Hip-Hop	0,000	0,075	0,062	0,062	0,512	0,288	0,000
R'n'B & S.	0,040	0,040	0,180	0,120	0,180	0,400	0,040
Sonstiges	0,134	0,075	0,148	0,250	0,085	0,210	0,097

Tabelle 3.12: Genrekonfusionsmatrix der Klassifikation mit der vollständigen Merkmalsmenge und $k = 8$ auf der CDs-Datenbank. Die Reihen stehen dabei für die tatsächlichen Genres der Stücke und die Spalten für die klassifizierten Genres. Jedes Stück wird dabei zu dem Genre klassifiziert, zu dem es den höchsten Zugehörigkeitsgrad hat, wenn dieser mindestens 0,5 ist.

Wenn man die balancierten relativen Fehler der beiden Klassifikationsmethoden vergleicht (s. Tabellen 3.13 und 3.14), sieht man, dass unsere Verbesserungsidee die Ergebnisse der Klassifikation verschlechtert hat. Nur für das Genre Pop/Rock der CDs-Datenbanken konnte eine Verbesserung erzielt werden. Diese schlechten Ergebnisse liegen vermutlich daran, dass innerhalb der Klassifikation eines Genres falsche Positive zwar geringere Zugehörigkeitsgrade haben als wahre Positive, dieser Vergleich aber nicht immer funktioniert, wenn man mehrere Genres gleichzeitig betrachtet.

CDs						
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B
Normal	0,116	0,294	0,239	0,343	0,130	0,476
Nur ein Genre pro Stück	0,154	0,429	0,424	0,274	0,180	0,515

Tabelle 3.13: Vergleich des m_{BRE} bei normaler binärer Klassifikation für jedes Genre einzeln und bei Klassifikation jedes Stücks in höchstens ein Genre auf der CDs-Datenbank für die vollständige Merkmalsmenge und $k = 3$.

1517						
	Classical	Electr. & D.	Jazz	Pop/Rock	Hip-Hop	R'n'B & S.
Normal	0,246	0,272	0,337	0,386	0,178	0,372
Nur ein Genre pro Stück	0,308	0,350	0,365	0,396	0,291	0,402

Tabelle 3.14: Vergleich des m_{BRE} bei normaler binärer Klassifikation für jedes Genre einzeln und bei Klassifikation jedes Stücks in höchstens ein Genre auf der 1517-Datenbank für die vollständige Merkmalsmenge und $k = 3$.

3.2.3 Relief-Merkmalsauswahl

Als letztes untersuchen wir nun die Merkmalsauswahl mit Relief. Wir vergleichen dazu den balancierten relativen Fehler der Klassifikation mit der vollständigen Merkmalsmenge und der Relief-Auswahl, bei der ein Teil dieser Merkmalsmenge für die Klassifikation ausgewählt wird.

Auf der CDs-Datenbank hat die Merkmalsauswahl mit Relief die Klassifikation insgesamt etwas verbessert (s. Tabelle 3.15). Bei dem Genre R'n'B konnte der Klassifikationsfehler deutlich verringert werden. Bei den Genres Electronic und Rap hat sich die Klassifikation allerdings durch die Relief-Merkmalsauswahl sogar verschlechtert.

Auf der 1517-Datenbank hat die Merkmalsauswahl die Klassifikationsergebnisse verschlechtert. Nur für die Genres Electronic and Dance und Pop/Rock haben sich die Ergebnisse der Klassifikation verbessert.

CDs							
	Classical	Electronic	Jazz	Pop/Rock	Rap	R'n'B	Gesamt
Vollst. Merkmalsmenge	0,102	0,289	0,244	0,338	0,139	0,477	0,265
Relief-Merkmalsausw.	0,093	0,295	0,230	0,337	0,183	0,394	0,256

Tabelle 3.15: Balancierter relativer Fehler für die vollständige Merkmalsmenge und die durch Relief ausgewählten Merkmale für die verschiedenen Genres der CDs-Datenbank.

1517							
	Classical	Electr. & D.	Jazz	Pop/Rock	Hip-Hop	R'n'B & S.	Gesamt
Vollst. Merkmalsmenge	0,234	0,279	0,331	0,399	0,184	0,360	0,298
Relief-Merkmalsausw.	0,239	0,276	0,362	0,373	0,214	0,394	0,310

Tabelle 3.16: Balancierter relativer Fehler für die vollständige Merkmalsmenge und die durch Relief ausgewählten Merkmale für die verschiedenen Genres der 1517-Datenbank.

Auch, wenn die Merkmalsauswahl nicht besonders gut funktioniert hat, ist es interessant sich anzuschauen, welche Merkmale von Relief als besonders gut, bzw. schlecht bewertet werden. Für zwei der verwendeten Merkmalsgruppen, wird die strukturelle Komplexität für drei verschiedene Fenstergrößen berechnet. Für die restlichen sechs Merkmalsgruppen wird die strukturelle Komplexität für sechs verschiedene Fenstergrößen berechnet (s. Abschnitt 3.1). Außerdem ist die strukturelle Komplexität, so wie wir sie hier verwenden, für jede Fenstergröße sieben-dimensional (s. Abschnitt 2.2.5). Es gibt also für jede Trainingsmenge, auf der wir Relief anwenden $(6 \cdot 6 + 2 \cdot 3) \cdot 7 = 294$ Merkmalsdimensionen, aus denen ausgewählt werden muss. Um zu untersuchen, welche Merkmale von Relief gut bewertet werden, sortieren wir diese 294 Merkmalsdimensionen absteigend nach der Relief-Varianz. Dann teilen wir diese 294 Ränge in sieben gleichgroße Teile, die wir im Folgenden *Bins* nennen, und zählen wir häufig eine Art von Merkmal auf einem Rang in diesen sieben Bins landet. Ein Merkmal, das von Relief als gut bewertet wird, landet dann häufiger in den vorderen Bins und ein schlechter bewertetes Merkmal landet häufiger in den hinteren Bins. Im Folgenden schauen wir uns an, in welchen Rang-Bins die strukturellen Komplexitätsmerkmale der verschiedenen Merkmalsgruppen für die Relief-Auswahl der verschiedenen Genres landen.

Es fällt dabei auf, dass auf beiden Datenbanken für klassische Musik die Merkmalsgruppe Chroma von Relief am besten bewertet wird (s. Abbildung 3.15). Auch besser als andere Merkmalsgruppen, wie z.B. Harmonie (s. Abbildung 3.16), die dieses Genre deutlich besser klassifizieren konnten, zumindest als sie einzeln getestet wurden (s. Tabellen 3.1 und 3.2).

Für Jazz wurde auf beiden Datenbanken die Merkmalsgruppe Akkordvektor von Relief gut bewertet (s. Abbildung 3.17), die auch relativ gut bei der Klassifikation dieses

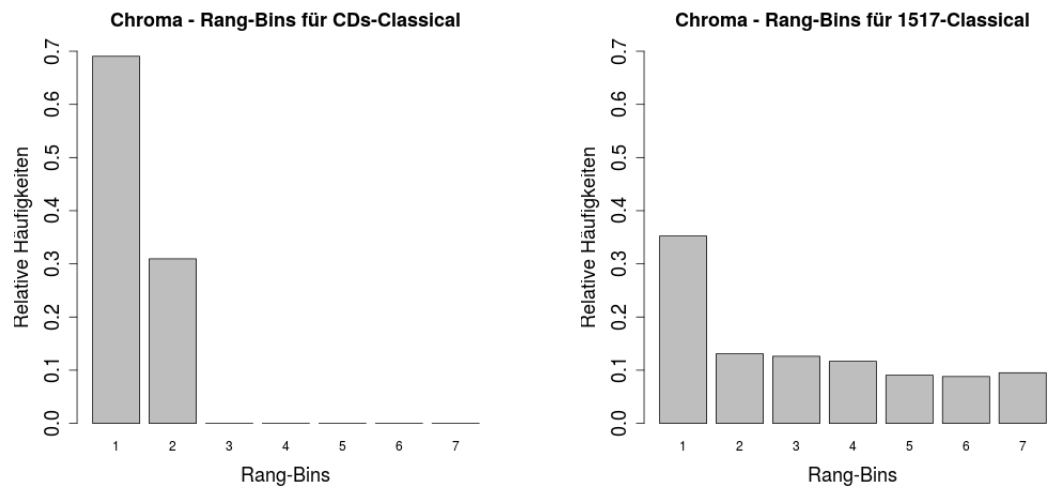


Abbildung 3.15: Relief-Ränge der Merkmalsgruppe Chroma für die Genres Classical (CDs) und Classical (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Chroma-Merkmale, die in dem jeweiligen Bin liegen.

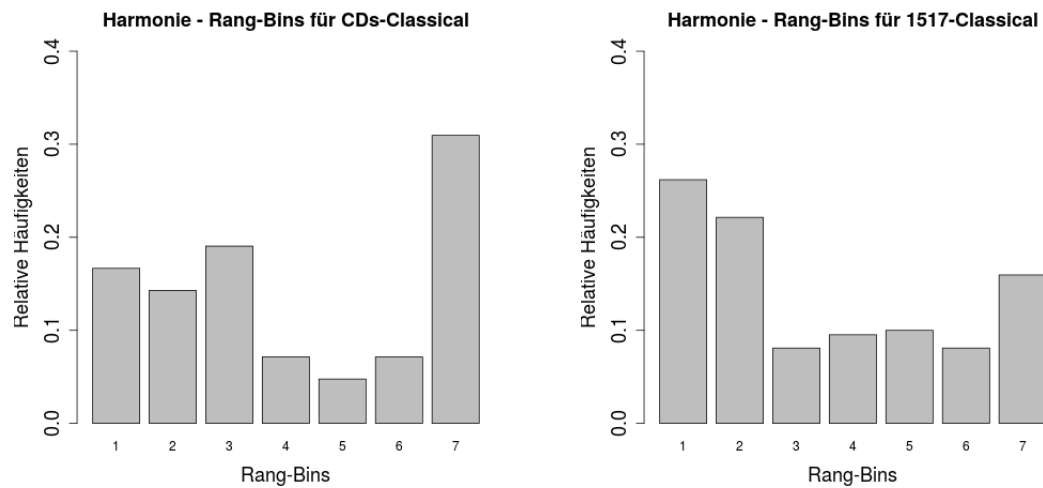


Abbildung 3.16: Relief-Ränge der Merkmalsgruppe Harmonie für die Genres Classical (CDs) und Classical (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Harmonie-Merkmale, die in dem jeweiligen Bin liegen.

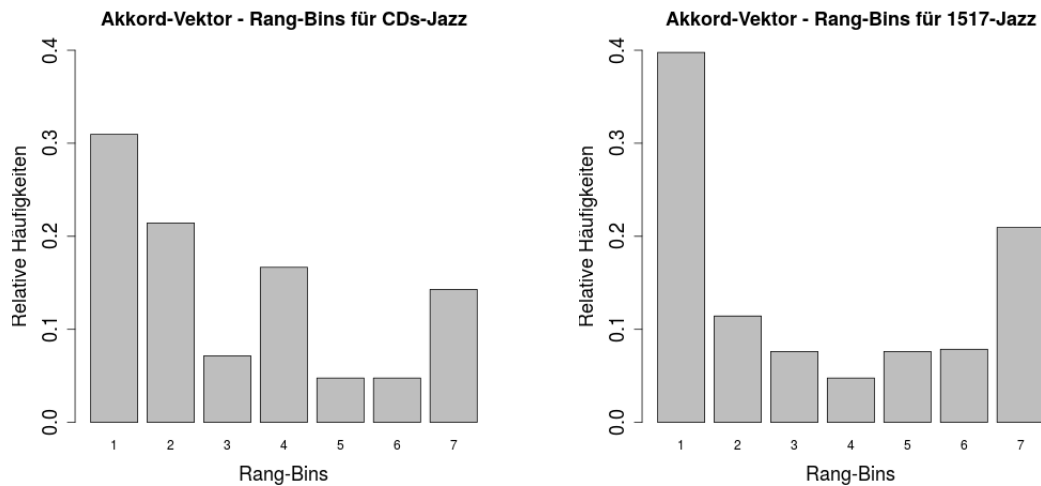


Abbildung 3.17: Relief-Ränge der Merkmalsgruppe Akkordvektor für die Genres Jazz (CDs) und Jazz (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Akkordvektor-Merkmale, die in dem jeweiligen Bin liegen.

Genres abschnitt (s. Tabellen 3.1 und 3.2). Trotzdem hat auf der 1517-Datenbank die Merkmalsauswahl der Klassifikation von Jazz geschadet.

Auffällig ist, dass obwohl die Merkmalsgruppe Chroma bei der Klassifikation des Genres Rap (CDs) äußerst gut war, sie nicht sehr gut von Relief bewertet wurde und im Gegensatz dazu die Merkmalsgruppe Instrumente, die für dieses Genre eher schlecht abschnitt, sehr gut bewertet wurde (s. Abbildung 3.18 und Tabelle 3.1). Das könnte erklären, warum sich die Klassifikation dieser beiden Genres durch Relief deutlich verschlechtert hat. Das Genre Hip-Hop der 1517-Datenbank, das ebenfalls sehr gut mit Chroma klassifiziert werden konnte, hat dieses Problem nicht (s. Abbildung 3.19). Trotzdem hat sich auch die Klassifikation dieses Genres durch die Relief-Merkmalsauswahl deutlich verschlechtert.

Für das Genre R'n'B der CDs-Datenbank wurden interessanterweise die Merkmalsgruppen Chroma und Klangfarbe durch Relief hoch bewertet (s. Abbildung 3.20), obwohl diese alleine zu eher schlechten Ergebnissen auf diesem Genre führten (s. Tabelle 3.1). Durch diese Auswahl konnte Relief dennoch die Klassifikation dieses Genres zumindest im Vergleich zur vollständigen Merkmalsmenge deutlich verbessern. Für einige einzelne Merkmalsgruppen, wie z.B. Chroma-verwandt, war die Klassifikation aber trotzdem besser (s. Tabelle 3.1).

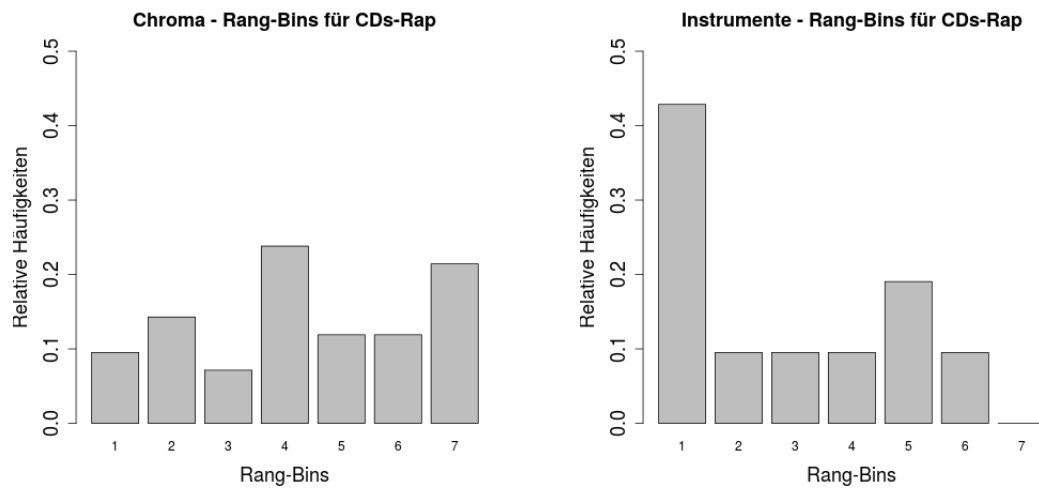


Abbildung 3.18: Relief-Ränge der Merkmalsgruppen Chroma und Instrumente für das Genre Rap (CDs). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Merkmale, die in dem jeweiligen Bin liegen.

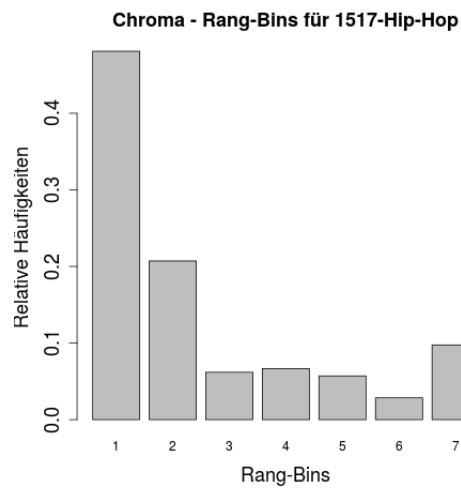


Abbildung 3.19: Relief-Ränge der Merkmalsgruppe Chroma und Hip-Hop (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Hip-Hop-Merkmale, die in dem jeweiligen Bin liegen.

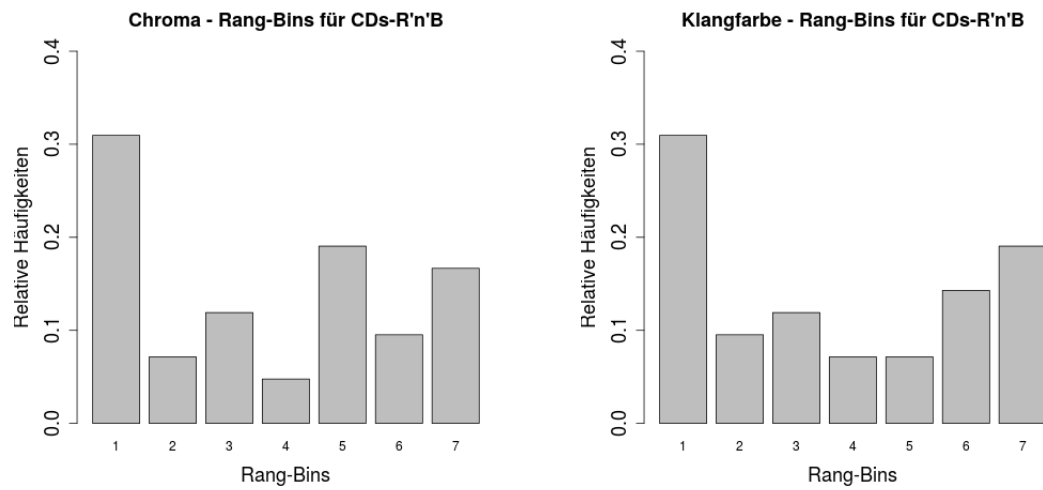


Abbildung 3.20: Relief-Ränge der Merkmalsgruppen Chroma und Klangfarbe für das Genre R'n'B (CDs). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Merkmale, die in dem jeweiligen Bin liegen.

Kapitel 4

Zusammenfassung und Ausblick

Durch die für diese Arbeit durchgeführten Experimente konnten wir einige Erkenntnisse darüber sammeln, wie man strukturelle Komplexitätsmerkmale zur Klassifikation von Musikgenres einsetzen kann.

So hängt die Komplexität der Merkmalsgruppen häufig auch mit der empfundenen Komplexität der Genres zusammen, besonders bei Merkmalsgruppen, die für Menschen leicht verständliche Eigenschaften, wie z.B. Harmonie oder Instrumente beschreiben. So lässt sich zum Beispiel klassische Musik auf beiden Datenbanken sehr gut mit der Merkmalsgruppe Harmonie klassifizieren und hat eine deutlich höhere Komplexität als die anderen Genres (s. Abschnitt 3.2.1). Als Mensch würde man auch Vermuten, dass klassische Musik harmonisch sehr komplex im Vergleich zu den anderen Genres ist. Auf der CDs-Datenbank ließ sich klassische Musik außerdem sehr gut mit der Merkmalsgruppe Instrumente klassifizieren, für die sie eine deutlich höhere Komplexität als die anderen Genres hat (s. Abschnitt 3.2.1). Dies hängt vermutlich damit zusammen, dass in klassischer Musik häufig viele verschiedene Instrumente vorkommen. Diese Instrumente stammen oft aus einer der Instrumentengruppen Blasinstrumente, Streichinstrumente, Klavier oder Gitarre, die durch diese Merkmalsgruppe erkannt werden (s. Abschnitt 2.1). Neben der Merkmalsgruppe Chroma, die für die meisten Genres recht gute Ergebnisse erreichte, ließ sich elektronische Musik auf beiden Datenbanken recht gut mit den Merkmalsgruppen Tempo und Rhythmus und Akkordvektor klassifizieren (s. Abschnitt 3.2.1). Für die Merkmalsgruppe Tempo und Rhythmus hat dieses Genre eine niedrige Komplexität für kleine Fenstergrößen und eine recht hohe Komplexität für große Fenstergrößen. Das lässt sich möglicherweise dadurch erklären, dass elektronische Musik häufig gleichbleibende, monotone Rhythmen hat, die sich dann aber zwischen verschiedenen längeren Abschnitten stark ändern können. Für das Merkmal Akkordvektor hat dieses Genre eine recht niedrige Komplexität, die sich für die verschiedenen Fenstergrößen kaum ändert, was sich dadurch erklären lassen könnte, dass elektronische Musik häufig sich immer wieder wiederholende, gleichbleibende Akkordfolgen besitzt. Das Genre Jazz lässt sich auf beiden Datenbanken

recht gut mit dem Merkmal Akkordvektor klassifizieren, für das es eine recht hohe Komplexität hat, was möglicherweise mit den recht komplexen Akkorden und Akkordfolgen zusammenhängt, die im Jazz häufig verwendet werden. Die Genres Rap (CDs) und Hip-Hop (1517) lassen sich sehr gut mit der Merkmalsmenge Chroma klassifizieren für die sie eine sehr geringe Komplexität haben (s. Abschnitt 3.2.1). Das lässt sich dadurch erklären, dass in Rap und Hip-Hop häufig ein gleichbleibender Beat über das gesamte Stück verwendet wird. Dadurch ändern sich die verwendeten Töne, die man bei Chroma erkennen kann, kaum über die Zeit.

Insgesamt lässt sich auf der CDs-Datenbank die Genres Classical und Rap und auf der 1517-Datenbank die Genres Classical und Hip-Hop am besten klassifizieren.

Bei unseren Experimenten führte strukturelle Komplexität meistens mit kleinen Fenstergrößen zu den besten Ergebnissen. Dies lässt sich dadurch erklären, dass bei großen Fenstergrößen die Merkmale der Musik über große Zeiträume hinweg zusammengefasst werden und die Stücke vor der Klassifikation in große Partitionen eingeteilt werden. In [20] zeigte sich, dass das besonders bei schwierigen Klassifikationsproblemen die Qualität der Klassifikation deutlich verschlechtert. Trotzdem könnten strukturelle Komplexitätsmerkmale großer Zeitfenster die Klassifikation möglicherweise verbessern, wenn man sie mit strukturellen Komplexitäten kleinerer Zeitfenster oder anderen Merkmalen kombiniert. Dies wäre eine interessante Fragestellung für zukünftige Arbeiten, da wir hier die verschiedenen Einstellungen der strukturellen Komplexität zum größten Teil nur einzeln betrachtet haben. Bei der Klassifikation mit der vollständigen Merkmalsmenge ließ sich aber bereits erkennen, dass die Kombination mehrerer unterschiedlicher Merkmale die Klassifikation deutlich verbessern kann (s. Abschnitt 3.2.1). Es wäre aber interessant, die verschiedenen möglichen Kombinationen genauer zu untersuchen.

Zwei Aspekte der strukturellen Komplexität, die wir in dieser Arbeit gar nicht untersucht haben sind die Zusammenfassungsfunktion und das Abstandsmaß, die zur Berechnung der Komplexität verwendet werden (s. Abschnitt 2.2.4). Es wäre interessant, ein Abstandsmaß zu verwenden, das eine musikalische Bedeutung hat, wie z.B. die in [3] beschriebene Methode, die den Abstand zweier Akkorde auf eine Weise bestimmt, die damit zusammenhängt wie unterschiedlich die beiden Akkorde musiktheoretisch gesehen werden. Um dieses Abstandsmaß zu verwenden, muss man allerdings eine Möglichkeit finden, wie man mehrere Akkorde so zusammenfassen kann, dass man dieses Abstandsmaß trotzdem anwenden kann, da es eigentlich nur dazu gedacht ist, den Abstand zweier Akkorde zu berechnen. Denkbar wäre es dafür z.B. den Modus als Zusammenfassungsfunktion zu verwenden.

Bei unseren Untersuchungen zum FKNN-Algorithmus zeigte sich, dass größere Werte für den Parameter k , der die Anzahl der verwendeten nächsten Nachbarn beschreibt, die Klassifikation tendenziell verbessern. Er scheint allerdings keinen sehr starken Einfluss auf die Klassifikation zu haben (s. Abschnitt 3.2.2). Das lässt sich dadurch erklären, dass die

Nachbarn, die durch ein höheres k zur Klassifikation hinzukommen nur einen geringen Einfluss haben, da sie weiter entfernt sind. Es wäre daher interessant den Parameter μ näher zu untersuchen, der bestimmt, wie stark der Einfluss der Nachbarn mit deren Entfernung abnimmt (s. Abschnitt 2.3.2). Diesen Parameter haben wir in dieser Arbeit gar nicht untersucht. Die Zugehörigkeitsgrade, die FKNN zur Klassifikation der Stücke berechnet, scheinen widerzuspiegeln, wie sicher man sich bei der Klassifikation eines Stückes sein kann. Wir haben getestet, ob es die Klassifikation verbessert, jedes Stück nur in das Genre zu klassifizieren, zu dem es den höchsten Zugehörigkeitsgrad hat. Leider verschlechterte dies die Klassifikation. Es wäre interessant andere Methoden zu untersuchen, wie man diese Zugehörigkeitsgrade nutzen kann, um bessere Ergebnisse zu erzielen, oder den Algorithmus in größere Systeme einzubauen.

Die Merkmalsauswahl mit Relief funktionierte leider nicht besonders gut. In manchen Situationen konnte sie die Klassifikation verbessern. Meistens hatte sie aber nur einen kleinen Einfluss oder verschlechterte die Klassifikation sogar (s. Abschnitt 3.2.3). Der Nachteil dieser Methode ist, dass sie die Auswirkung der Merkmalsauswahl auf die tatsächliche Klassifikation des Algorithmus, nicht mit in die Entscheidung einbezieht, da sie eine Filtermethode ist (s. Abschnitt 2.2.6). Außerdem werden die Zusammenhänge zwischen den verschiedenen Merkmalen bei der Auswahl kaum berücksichtigt. Deswegen wäre es interessant andere Merkmalsauswahlmethoden, wie z.B. evolutionäre Merkmalsauswahl oder sogar mehrkriterielle Merkmalsauswahl, wie sie in [16] eingesetzt wurde, gezielt auf strukturellen Komplexitätsmerkmalen zu testen um ein besseres Verständnis dafür zu erlangen, welche Kombinationen aus Merkmalsgruppen und Fenstergrößen gut funktionieren.

Anhang A

Anhang

A.1 Merkmalsgruppen

Tabelle A.1: Zusammensetzung der verwendeten Merkmalgruppen für die strukturelle Komplexität mit Referenzen für weitere Informationen über die Merkmale und deren im AMUSE-Framework verwendeten IDs [21].

Name	Referenz	ID
Akkorde		
Number of different chords in 10s	[16]	257
Number of chord changes in 10s	[16]	258
Number of the most frequent 20, 40 and 60 percent of chords with regard to their duration	[16]	259
Chroma		
Bass chroma	[11]	251
Chroma	[11]	250
Chroma-verwandt		
Amplitude, position and width of the 1st spectral peak	[15]	211
Amplitude position and width of the 2nd spectral peak	[15]	212
Amplitude, position and width of the 3rd spectral peak	[15]	213
amplitude, position and width of the 4th spectral peak	[15]	214
Amplitude, position and width of the 5th spectral peak	[15]	215
Chroma maximum	[15]	205
Chroma tone with the maximum strength	[15]	207
Fundamental frequency	[15]	200
Inharmonicity	[10]	201
Harmonic change detection function	[10]	217

Harmonie		
Chroma maximum	[15]	205
Chroma tone with the maximum strength	[15]	207
Consonance	[11]	255
Harmonic change	[11]	254
Harmonic change detection function	[10]	17
Interval strengths estimated from 10 highest semitone values	[16]	260
Interval strengths estimated from the semitone spectrum above 3/4 of the maximum value	[16]	261
Key and its clarity	[10]	202
Local tuning	[11]	253
Major/minor alignment	[10]	203
Number of different chords in 10s	[16]	257
Number of chord changes in 10s	[16]	258
Shares of the most frequent 20, 40 and 60 percent of chords with regard to their duration	[16]	259
Strength of major keys	[10]	209
Strength of minor keys	[10]	210
Tonal centroid vector	[10]	216
Instrumente		
Guitar RF	[16]	2001
Guitar SVM	[16]	2003
Piano RF	[16]	2021
Piano SVM	[16]	2023
Wind RF	[16]	2041
Wind SVM	[16]	2043
Strings RF	[16]	2061
Strings SVM	[16]	2063
Tempo und Rhythmus		
Estimated beat number per minute	[15]	421
Estimated tatum number per minute	[15]	422
Estimated onset number per minute	[15]	420
Tempo based on onset times	[10]	425
Five peaks of fluctuation curves summed across all bands	[10]	427
Characteristics of fluctuation patterns	[15]	410
Rhythmic clarity	[10]	418

Klangfarbe		
Low energy	[15]	6
Root mean square	[15]	4
RMS peak number in 3 seconds	[10]	11
RMS peak number above half of maximum peak in 3 seconds	[10]	12
Zero-crossing rate	[15]	0
Average distance between extremal spectral values and its variance	[15]	2
Average distance between zero-crossings of the time-domain signal and its variance	[15]	3
Normalised energy of harmonic components	[15]	7
Spectral bandwidth	[15]	16
Spectral brightness	[10]	23
Spectral centroid	[15]	14
Spectral crest factor	[15]	19
Spectral discrepancy	[15]	31
Spectral extend	[15]	21
Spectral flatness measure	[15]	20
Spectral flux	[15]	22
Spectral irregularity	[10]	15
Spectral kurtosis	[15]	18
Spectral skewness	[15]	17
Spectral slope	[15]	29
Sensory roughness	[10]	24
Sub-band energy ratio	[15]	25
Tristimulus	[15]	10
y-axis intercept	[15]	30
Delta MFCCs	[10]	48
Mel frequency cepstral coefficients	[15]	39
Angles in phase domain	[15]	32
Distances in phase domain	[15]	33

A.2 Balancierter relativer Fehler

Tabelle A.2: Ergebnisse der Klassifikation auf der CDs-Datenbank. In der Tabelle findet man den balancierten relativen Fehler für alle Kombinationen aus Genres, Fenstergrößen W , nächsten Nachbarn k und verwendeten Merkmalsgruppen. Außerdem findet man die Ergebnisse der Klassifikation mit der vollständigen Merkmalsmenge und der durch die Relief-Merkmauswahl ausgewählten Merkmalsmenge.

Classical				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,257	0,269	0,223	0,259
4s	0,218	0,233	0,216	0,216
8s	0,158	0,163	0,158	0,161
16s	0,173	0,192	0,194	0,153
32s	0,170	0,161	0,156	0,153
64s	0,167	0,167	0,159	0,202
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,096	0,091	0,115	0,115
4s	0,134	0,105	0,084	0,081
8s	0,144	0,108	0,103	0,101
16s	0,290	0,302	0,300	0,302
32s	0,403	0,351	0,365	0,365
64s	0,509	0,477	0,477	0,497
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,120	0,120	0,115	0,110
4s	0,081	0,108	0,108	0,089
8s	0,139	0,117	0,110	0,101
16s	0,252	0,242	0,237	0,232
32s	0,247	0,218	0,223	0,225
64s	0,377	0,301	0,272	0,275

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,324	0,295	0,343	0,379
4s	0,304	0,271	0,271	0,223
8s	0,144	0,144	0,132	0,132
16s	0,357	0,278	0,271	0,242
32s	0,371	0,412	0,367	0,367
64s	0,431	0,339	0,364	0,356
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,194	0,158	0,165	0,160
4s	0,213	0,180	0,177	0,192
8s	0,273	0,290	0,285	0,295
16s	0,371	0,314	0,357	0,321
32s	0,299	0,295	0,287	0,290
64s	0,429	0,396	0,394	0,368
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,374	0,324	0,329	0,331
4s	0,338	0,312	0,355	0,381
8s	0,271	0,271	0,247	0,276
16s	0,333	0,372	0,367	0,362
32s	0,256	0,230	0,256	0,266
64s	0,375	0,377	0,352	0,352
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,098	0,084	0,093	0,091
32s	0,149	0,146	0,146	0,149
64s	0,213	0,194	0,189	0,194
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,463	0,407	0,429	0,414
32s	0,511	0,470	0,472	0,438
64s	0,516	0,495	0,512	0,485

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,081	0,098	0,114	0,116
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,078	0,096	0,096	0,101
Electronic				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,307	0,296	0,312	0,310
4s	0,385	0,343	0,339	0,336
8s	0,305	0,283	0,300	0,302
16s	0,335	0,315	0,279	0,279
32s	0,430	0,444	0,429	0,426
64s	0,455	0,399	0,397	0,403
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,369	0,331	0,348	0,304
4s	0,441	0,432	0,432	0,415
8s	0,410	0,386	0,431	0,436
16s	0,350	0,368	0,325	0,327
32s	0,517	0,507	0,513	0,515
64s	0,578	0,505	0,510	0,498
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,398	0,409	0,435	0,430
4s	0,393	0,426	0,409	0,409
8s	0,393	0,366	0,392	0,437
16s	0,462	0,472	0,460	0,455
32s	0,402	0,392	0,411	0,417
64s	0,502	0,537	0,543	0,536

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,335	0,325	0,322	0,337
4s	0,332	0,317	0,296	0,298
8s	0,349	0,354	0,315	0,315
16s	0,312	0,321	0,347	0,342
32s	0,422	0,367	0,340	0,314
64s	0,453	0,487	0,453	0,463
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,262	0,262	0,288	0,283
4s	0,324	0,308	0,300	0,312
8s	0,389	0,408	0,408	0,403
16s	0,431	0,366	0,385	0,385
32s	0,470	0,474	0,459	0,445
64s	0,472	0,432	0,401	0,403
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,365	0,327	0,280	0,287
4s	0,396	0,368	0,330	0,291
8s	0,342	0,269	0,290	0,269
16s	0,386	0,393	0,385	0,383
32s	0,423	0,423	0,401	0,420
64s	0,511	0,486	0,476	0,490
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,481	0,522	0,489	0,506
32s	0,487	0,462	0,457	0,466
64s	0,543	0,536	0,535	0,519
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,446	0,416	0,378	0,397
32s	0,459	0,454	0,453	0,477
64s	0,425	0,405	0,441	0,431

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,303	0,270	0,291	0,291
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,291	0,291	0,305	0,293
Jazz				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,365	0,346	0,298	0,322
4s	0,437	0,408	0,396	0,429
8s	0,289	0,280	0,263	0,289
16s	0,361	0,359	0,363	0,335
32s	0,490	0,439	0,425	0,437
64s	0,419	0,405	0,403	0,415
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,385	0,409	0,417	0,393
4s	0,408	0,406	0,382	0,370
8s	0,434	0,383	0,390	0,388
16s	0,421	0,469	0,460	0,417
32s	0,557	0,561	0,565	0,553
64s	0,352	0,442	0,460	0,452
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,327	0,339	0,344	0,330
4s	0,332	0,341	0,337	0,334
8s	0,220	0,220	0,222	0,235
16s	0,389	0,367	0,353	0,346
32s	0,429	0,417	0,456	0,449
64s	0,461	0,482	0,501	0,489

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,426	0,419	0,395	0,383
4s	0,402	0,364	0,364	0,364
8s	0,396	0,329	0,329	0,343
16s	0,495	0,502	0,481	0,459
32s	0,487	0,489	0,541	0,555
64s	0,425	0,383	0,388	0,390
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,387	0,366	0,418	0,385
4s	0,349	0,370	0,401	0,373
8s	0,502	0,476	0,485	0,447
16s	0,439	0,410	0,391	0,405
32s	0,520	0,518	0,551	0,558
64s	0,515	0,499	0,485	0,506
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,369	0,395	0,326	0,321
4s	0,318	0,344	0,361	0,358
8s	0,233	0,233	0,259	0,254
16s	0,319	0,292	0,318	0,318
32s	0,457	0,435	0,451	0,439
64s	0,382	0,423	0,411	0,415
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,297	0,251	0,228	0,233
32s	0,384	0,384	0,377	0,374
64s	0,414	0,412	0,397	0,392
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,440	0,397	0,429	0,412
32s	0,526	0,511	0,49	0,466
64s	0,495	0,483	0,449	0,451

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,263	0,255	0,217	0,241
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,213	0,227	0,237	0,244
	Pop/Rock			
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,353	0,337	0,341	0,348
4s	0,398	0,380	0,372	0,387
8s	0,419	0,452	0,416	0,421
16s	0,426	0,402	0,374	0,370
32s	0,434	0,444	0,436	0,454
64s	0,435	0,454	0,414	0,406
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,368	0,350	0,361	0,350
4s	0,376	0,370	0,373	0,376
8s	0,410	0,428	0,410	0,405
16s	0,448	0,428	0,407	0,411
32s	0,433	0,418	0,405	0,409
64s	0,424	0,453	0,436	0,422
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,393	0,382	0,391	0,391
4s	0,458	0,445	0,429	0,419
8s	0,484	0,471	0,448	0,454
16s	0,477	0,455	0,466	0,452
32s	0,420	0,409	0,396	0,419
64s	0,474	0,462	0,456	0,478

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,349	0,388	0,357	0,348
4s	0,411	0,392	0,375	0,396
8s	0,446	0,427	0,409	0,426
16s	0,456	0,451	0,443	0,447
32s	0,481	0,446	0,431	0,427
64s	0,469	0,497	0,502	0,481
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,472	0,444	0,452	0,459
4s	0,455	0,490	0,488	0,453
8s	0,461	0,437	0,484	0,479
16s	0,514	0,475	0,442	0,429
32s	0,497	0,468	0,442	0,436
64s	0,498	0,511	0,507	0,499
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,327	0,350	0,361	0,361
4s	0,359	0,375	0,363	0,359
8s	0,352	0,327	0,334	0,346
16s	0,374	0,387	0,387	0,382
32s	0,471	0,471	0,447	0,449
64s	0,427	0,418	0,418	0,394
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,427	0,426	0,416	0,414
32s	0,436	0,438	0,411	0,411
64s	0,405	0,416	0,419	0,417
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,497	0,508	0,503	0,515
32s	0,476	0,501	0,472	0,474
64s	0,491	0,468	0,465	0,450

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,332	0,341	0,337	0,343
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,339	0,342	0,340	0,329
Rap				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,142	0,138	0,143	0,145
4s	0,176	0,150	0,143	0,150
8s	0,159	0,159	0,142	0,125
16s	0,182	0,199	0,211	0,208
32s	0,270	0,246	0,241	0,241
64s	0,256	0,260	0,250	0,248
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,283	0,302	0,293	0,312
4s	0,342	0,371	0,344	0,363
8s	0,552	0,511	0,501	0,515
16s	0,453	0,498	0,488	0,493
32s	0,380	0,407	0,405	0,398
64s	0,379	0,476	0,450	0,462
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,423	0,442	0,442	0,437
4s	0,457	0,409	0,385	0,382
8s	0,453	0,470	0,465	0,460
16s	0,390	0,478	0,458	0,451
32s	0,496	0,450	0,459	0,485
64s	0,472	0,466	0,440	0,447

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,301	0,359	0,325	0,359
4s	0,365	0,384	0,367	0,379
8s	0,402	0,375	0,437	0,446
16s	0,446	0,422	0,400	0,393
32s	0,456	0,405	0,357	0,388
64s	0,329	0,344	0,356	0,356
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,489	0,482	0,489	0,486
4s	0,501	0,460	0,467	0,465
8s	0,471	0,498	0,474	0,450
16s	0,544	0,539	0,541	0,579
32s	0,504	0,463	0,491	0,489
64s	0,431	0,395	0,381	0,402
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,392	0,385	0,387	0,351
4s	0,461	0,408	0,412	0,426
8s	0,435	0,456	0,427	0,424
16s	0,401	0,437	0,404	0,406
32s	0,489	0,497	0,497	0,478
64s	0,463	0,472	0,455	0,503
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,483	0,485	0,445	0,461
32s	0,383	0,412	0,414	0,412
64s	0,347	0,367	0,364	0,347
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,484	0,517	0,467	0,485
32s	0,571	0,573	0,575	0,563
64s	0,503	0,517	0,528	0,526

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,161	0,142	0,125	0,128
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,171	0,195	0,193	0,174
R'n'B				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,463	0,465	0,474	0,446
4s	0,461	0,416	0,418	0,437
8s	0,385	0,439	0,401	0,432
16s	0,473	0,420	0,474	0,505
32s	0,500	0,456	0,470	0,482
64s	0,467	0,509	0,494	0,517
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,329	0,335	0,318	0,318
4s	0,360	0,357	0,364	0,381
8s	0,419	0,409	0,444	0,387
16s	0,366	0,380	0,416	0,394
32s	0,459	0,493	0,496	0,483
64s	0,488	0,464	0,500	0,497
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,407	0,395	0,340	0,374
4s	0,400	0,354	0,342	0,357
8s	0,469	0,487	0,515	0,508
16s	0,472	0,477	0,493	0,452
32s	0,450	0,381	0,376	0,374
64s	0,508	0,493	0,492	0,490

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,422	0,393	0,386	0,372
4s	0,360	0,405	0,383	0,372
8s	0,450	0,506	0,508	0,465
16s	0,529	0,478	0,447	0,445
32s	0,396	0,414	0,376	0,380
64s	0,485	0,497	0,468	0,496
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,509	0,490	0,509	0,484
4s	0,440	0,501	0,496	0,494
8s	0,523	0,508	0,527	0,515
16s	0,439	0,433	0,407	0,433
32s	0,448	0,428	0,461	0,451
64s	0,462	0,466	0,427	0,419
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,477	0,481	0,515	0,488
4s	0,492	0,476	0,430	0,447
8s	0,468	0,489	0,501	0,468
16s	0,482	0,527	0,519	0,528
32s	0,489	0,455	0,457	0,474
64s	0,391	0,400	0,387	0,421
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,433	0,459	0,485	0,427
32s	0,434	0,457	0,468	0,454
64s	0,524	0,509	0,487	0,446
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,535	0,476	0,471	0,492
32s	0,491	0,471	0,48	0,508
64s	0,522	0,538	0,526	0,506

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,486	0,476	0,474	0,474
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,380	0,404	0,409	0,385

Tabelle A.3: Ergebnisse der Klassifikation auf der 1517-Datenbank. In der Tabelle findet man den balancierten relativen Fehler für alle Kombinationen aus Genres, Fenstergrößen W , nächsten Nachbarn k und verwendeten Merkmalsgruppen. Außerdem findet man die Ergebnisse der Klassifikation mit der vollständigen Merkmalsmenge und der durch die Relief-Merkmalsauswahl ausgewählten Merkmalsmenge.

Classical				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,347	0,308	0,308	0,322
4s	0,331	0,302	0,289	0,290
8s	0,265	0,248	0,213	0,223
16s	0,296	0,307	0,294	0,291
32s	0,313	0,286	0,286	0,287
64s	0,281	0,266	0,266	0,261
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,277	0,268	0,253	0,262
4s	0,306	0,288	0,290	0,283
8s	0,329	0,298	0,277	0,275
16s	0,380	0,369	0,381	0,382
32s	0,412	0,392	0,397	0,376
64s	0,499	0,519	0,464	0,468
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,201	0,210	0,205	0,204
4s	0,207	0,217	0,205	0,200
8s	0,197	0,197	0,193	0,179
16s	0,290	0,311	0,321	0,309
32s	0,382	0,377	0,360	0,364
64s	0,338	0,311	0,319	0,316

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,460	0,458	0,468	0,468
4s	0,408	0,415	0,398	0,413
8s	0,274	0,282	0,278	0,279
16s	0,376	0,368	0,370	0,357
32s	0,480	0,441	0,431	0,435
64s	0,437	0,420	0,407	0,406
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,302	0,304	0,301	0,304
4s	0,326	0,338	0,313	0,329
8s	0,345	0,342	0,357	0,337
16s	0,378	0,390	0,365	0,394
32s	0,448	0,486	0,449	0,423
64s	0,493	0,424	0,438	0,431
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,352	0,380	0,348	0,359
4s	0,470	0,419	0,395	0,391
8s	0,385	0,398	0,402	0,394
16s	0,439	0,416	0,430	0,410
32s	0,437	0,437	0,442	0,442
64s	0,400	0,380	0,384	0,384
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,258	0,263	0,257	0,272
32s	0,314	0,291	0,320	0,302
64s	0,373	0,350	0,347	0,357
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,444	0,430	0,436	0,418
32s	0,416	0,441	0,452	0,404
64s	0,504	0,464	0,500	0,463

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,228	0,219	0,245	0,246
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,253	0,245	0,229	0,230
	Electronic and Dance			
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,272	0,263	0,256	0,258
4s	0,304	0,296	0,295	0,289
8s	0,321	0,322	0,308	0,290
16s	0,371	0,319	0,322	0,327
32s	0,448	0,445	0,460	0,428
64s	0,440	0,432	0,431	0,426
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,412	0,402	0,405	0,399
4s	0,389	0,378	0,391	0,375
8s	0,441	0,448	0,437	0,445
16s	0,391	0,367	0,372	0,366
32s	0,408	0,397	0,409	0,395
64s	0,429	0,389	0,380	0,366
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,308	0,309	0,306	0,299
4s	0,330	0,308	0,303	0,310
8s	0,409	0,411	0,390	0,400
16s	0,377	0,365	0,347	0,347
32s	0,372	0,353	0,364	0,334
64s	0,425	0,402	0,377	0,377

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,337	0,349	0,334	0,344
4s	0,356	0,325	0,302	0,300
8s	0,341	0,339	0,344	0,342
16s	0,380	0,327	0,342	0,336
32s	0,429	0,398	0,373	0,363
64s	0,445	0,401	0,396	0,388
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,325	0,301	0,312	0,313
4s	0,313	0,312	0,310	0,300
8s	0,337	0,339	0,313	0,300
16s	0,319	0,281	0,269	0,270
32s	0,400	0,367	0,366	0,362
64s	0,366	0,369	0,370	0,355
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,307	0,319	0,316	0,314
4s	0,276	0,290	0,288	0,279
8s	0,344	0,338	0,339	0,338
16s	0,359	0,355	0,348	0,354
32s	0,441	0,385	0,394	0,400
64s	0,439	0,428	0,410	0,418
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,447	0,458	0,458	0,450
32s	0,443	0,470	0,475	0,474
64s	0,465	0,443	0,439	0,427
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,451	0,417	0,400	0,411
32s	0,411	0,401	0,398	0,391
64s	0,405	0,387	0,374	0,379

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,289	0,280	0,277	0,272
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,291	0,277	0,273	0,262
Jazz				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,460	0,455	0,437	0,448
4s	0,435	0,431	0,438	0,426
8s	0,354	0,373	0,365	0,356
16s	0,447	0,429	0,430	0,434
32s	0,448	0,448	0,431	0,427
64s	0,473	0,439	0,436	0,425
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,483	0,471	0,434	0,442
4s	0,417	0,441	0,439	0,427
8s	0,477	0,467	0,465	0,458
16s	0,510	0,513	0,492	0,493
32s	0,466	0,475	0,448	0,458
64s	0,404	0,369	0,379	0,381
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,449	0,442	0,462	0,461
4s	0,466	0,463	0,443	0,466
8s	0,441	0,435	0,438	0,419
16s	0,467	0,449	0,431	0,441
32s	0,467	0,418	0,425	0,433
64s	0,521	0,509	0,498	0,500

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,435	0,481	0,483	0,476
4s	0,474	0,464	0,450	0,450
8s	0,419	0,423	0,419	0,420
16s	0,495	0,509	0,514	0,517
32s	0,485	0,488	0,477	0,486
64s	0,453	0,435	0,427	0,437
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,518	0,530	0,523	0,496
4s	0,474	0,443	0,459	0,448
8s	0,481	0,459	0,463	0,480
16s	0,515	0,508	0,492	0,489
32s	0,495	0,498	0,488	0,482
64s	0,434	0,453	0,445	0,436
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,418	0,399	0,377	0,389
4s	0,403	0,398	0,392	0,389
8s	0,430	0,405	0,398	0,398
16s	0,408	0,373	0,378	0,371
32s	0,414	0,413	0,424	0,413
64s	0,445	0,442	0,428	0,421
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,367	0,374	0,370	0,363
32s	0,470	0,417	0,400	0,398
64s	0,457	0,439	0,433	0,433
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,487	0,481	0,463	0,452
32s	0,487	0,433	0,438	0,441
64s	0,516	0,532	0,510	0,513

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,326	0,332	0,330	0,337
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,366	0,371	0,358	0,355
	Pop/Rock			
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,463	0,451	0,439	0,467
4s	0,481	0,521	0,516	0,504
8s	0,512	0,509	0,514	0,516
16s	0,438	0,415	0,413	0,385
32s	0,407	0,408	0,413	0,407
64s	0,491	0,502	0,496	0,486
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,339	0,312	0,312	0,319
4s	0,383	0,365	0,369	0,350
8s	0,364	0,340	0,357	0,349
16s	0,493	0,503	0,486	0,485
32s	0,463	0,476	0,464	0,478
64s	0,524	0,523	0,513	0,519
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,458	0,435	0,434	0,451
4s	0,474	0,458	0,481	0,470
8s	0,476	0,485	0,468	0,467
16s	0,496	0,493	0,507	0,506
32s	0,494	0,500	0,511	0,485
64s	0,447	0,441	0,456	0,446

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,359	0,348	0,340	0,344
4s	0,371	0,356	0,346	0,352
8s	0,377	0,365	0,350	0,363
16s	0,484	0,461	0,468	0,471
32s	0,486	0,507	0,508	0,505
64s	0,508	0,495	0,481	0,488
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,526	0,512	0,495	0,504
4s	0,508	0,505	0,502	0,491
8s	0,475	0,487	0,466	0,454
16s	0,514	0,501	0,497	0,477
32s	0,493	0,441	0,448	0,464
64s	0,497	0,521	0,491	0,487
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,493	0,486	0,479	0,489
4s	0,486	0,484	0,494	0,483
8s	0,500	0,516	0,515	0,537
16s	0,453	0,441	0,441	0,441
32s	0,446	0,462	0,450	0,448
64s	0,471	0,489	0,494	0,490
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,344	0,365	0,344	0,348
32s	0,435	0,442	0,426	0,429
64s	0,496	0,495	0,506	0,503
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,476	0,496	0,498	0,470
32s	0,505	0,526	0,528	0,547
64s	0,451	0,465	0,470	0,452

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,400	0,395	0,413	0,386
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,381	0,384	0,372	0,356
Hip-Hop				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,249	0,236	0,235	0,236
4s	0,195	0,188	0,187	0,189
8s	0,200	0,205	0,196	0,197
16s	0,288	0,270	0,272	0,281
32s	0,345	0,322	0,326	0,344
64s	0,388	0,324	0,311	0,299
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,323	0,338	0,342	0,339
4s	0,359	0,361	0,373	0,373
8s	0,454	0,413	0,400	0,414
16s	0,419	0,433	0,418	0,414
32s	0,518	0,487	0,470	0,466
64s	0,390	0,383	0,385	0,378
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,297	0,284	0,270	0,278
4s	0,283	0,272	0,262	0,267
8s	0,342	0,333	0,328	0,303
16s	0,351	0,358	0,369	0,349
32s	0,354	0,351	0,348	0,353
64s	0,464	0,520	0,501	0,517

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,425	0,416	0,396	0,398
4s	0,399	0,369	0,388	0,384
8s	0,393	0,364	0,380	0,346
16s	0,442	0,465	0,459	0,458
32s	0,427	0,386	0,407	0,401
64s	0,443	0,435	0,411	0,400
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,468	0,483	0,446	0,447
4s	0,491	0,488	0,498	0,494
8s	0,476	0,453	0,451	0,447
16s	0,398	0,406	0,391	0,411
32s	0,399	0,396	0,404	0,392
64s	0,406	0,382	0,382	0,386
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,303	0,293	0,286	0,306
4s	0,307	0,308	0,293	0,285
8s	0,320	0,333	0,322	0,327
16s	0,404	0,367	0,366	0,357
32s	0,424	0,420	0,420	0,413
64s	0,477	0,520	0,496	0,495
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,349	0,335	0,345	0,353
32s	0,427	0,405	0,407	0,393
64s	0,402	0,408	0,387	0,382
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,418	0,390	0,402	0,411
32s	0,440	0,430	0,455	0,442
64s	0,452	0,503	0,497	0,494

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,201	0,180	0,179	0,178
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,216	0,213	0,217	0,210
R'n'B and Soul				
	Chroma			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,415	0,409	0,407	0,404
4s	0,406	0,397	0,397	0,417
8s	0,369	0,366	0,363	0,362
16s	0,465	0,437	0,434	0,432
32s	0,464	0,440	0,429	0,436
64s	0,436	0,417	0,425	0,412
	Chroma-verwandt			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,377	0,355	0,359	0,359
4s	0,434	0,424	0,404	0,414
8s	0,443	0,425	0,418	0,420
16s	0,474	0,475	0,480	0,486
32s	0,462	0,449	0,418	0,405
64s	0,502	0,520	0,524	0,517
	Harmonie			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,398	0,401	0,409	0,408
4s	0,393	0,377	0,398	0,395
8s	0,463	0,417	0,433	0,434
16s	0,498	0,495	0,470	0,497
32s	0,459	0,441	0,450	0,456
64s	0,474	0,438	0,461	0,464

	Klangfarbe			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,416	0,411	0,420	0,431
4s	0,414	0,424	0,434	0,419
8s	0,428	0,449	0,460	0,417
16s	0,446	0,432	0,430	0,438
32s	0,491	0,436	0,446	0,455
64s	0,446	0,463	0,459	0,458
	Tempo und Rhythmus			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,500	0,478	0,477	0,489
4s	0,451	0,484	0,506	0,509
8s	0,465	0,464	0,451	0,456
16s	0,472	0,485	0,467	0,455
32s	0,464	0,477	0,461	0,472
64s	0,477	0,467	0,462	0,475
	Akkordvektor			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
2s	0,427	0,405	0,403	0,405
4s	0,486	0,503	0,469	0,477
8s	0,549	0,521	0,529	0,522
16s	0,507	0,495	0,486	0,503
32s	0,483	0,487	0,481	0,499
64s	0,529	0,526	0,540	0,533
	Instrumente			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,447	0,446	0,434	0,416
32s	0,443	0,413	0,427	0,443
64s	0,459	0,499	0,509	0,517
	Akkorde			
W	$k = 2$	$k = 4$	$k = 6$	$k = 8$
16s	0,493	0,482	0,474	0,500
32s	0,456	0,455	0,490	0,482
64s	0,468	0,468	0,485	0,480

	Vollständige Merkmalsmenge			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,356	0,354	0,357	0,372
	Relief-Merkmalsauswahl			
	$k = 2$	$k = 4$	$k = 6$	$k = 8$
	0,392	0,386	0,403	0,396

Abbildungsverzeichnis

2.1	Erzeugung der harmonisierten Merkmalsmatrix X^H [16, S. 34]	10
3.1	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Chroma auf der CDs- und der 1517-Datenbank.	30
3.2	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Chroma-verwandt auf der CDs- und der 1517-Datenbank.	30
3.3	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Harmonie auf der CDs- und der 1517-Datenbank.	31
3.4	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Klangfarbe auf der CDs- und der 1517-Datenbank.	31
3.5	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Tempo und Rhythmus auf der CDs- und der 1517-Datenbank.	32
3.6	Durchschnittliche strukturelle Komplexität des Merkmals Akkordvektor auf der CDs- und der 1517-Datenbank.	33
3.7	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Instrumente auf der CDs- und der 1517-Datenbank.	34
3.8	Durchschnittliche strukturelle Komplexität der Merkmalsgruppe Akkorde auf der CDs- und der 1517-Datenbank.	34
3.9	m_{BRE} für die Genres Electronic and Dance und Jazz der 1517-Datenbank für alle Werte von k und W und die Merkmalsgruppe Chroma-verwandt.	37
3.10	m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Chroma-verwandt.	37
3.11	m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Harmonie.	38
3.12	m_{BRE} für das Genre Classical der CDs-Datenbank (links) und das Genre Classical der 1517-Datenbank (rechts) für alle Werte von k und W und die Merkmalsgruppe Klangfarbe.	39

3.13	m_{BRE} für die Genres Electronic (CDs) und Electronic and Dance (1517) für alle Werte von k und W und die Merkmalsgruppe Tempo und Rhythmus.	39
3.14	m_{BRE} für die Genres Classical und Jazz der CDs-Datenbank für alle Werte von k und W und die Merkmalsgruppe Tempo und Rhythmus.	40
3.15	Relief-Ränge der Merkmalsgruppe Chroma für die Genres Classical (CDs) und Classical (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Chroma-Merkmale, die in dem jeweiligen Bin liegen.	48
3.16	Relief-Ränge der Merkmalsgruppe Harmonie für die Genres Classical (CDs) und Classical (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Harmonie-Merkmale, die in dem jeweiligen Bin liegen.	48
3.17	Relief-Ränge der Merkmalsgruppe Akkordvektor für die Genres Jazz (CDs) und Jazz (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Akkordvektor-Merkmale, die in dem jeweiligen Bin liegen.	49
3.18	Relief-Ränge der Merkmalsgruppen Chroma und Instrumente für das Genre Rap (CDs). Auf der x-Achse stehen die sieben Rang-Bins und auf der y- Achse der Anteil der Merkmale, die in dem jeweiligen Bin liegen.	50
3.19	Relief-Ränge der Merkmalsgruppe Chroma und Hip-Hop (1517). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Hip-Hop-Merkmale, die in dem jeweiligen Bin liegen.	50
3.20	Relief-Ränge der Merkmalsgruppen Chroma und Klangfarbe für das Genre R'n'B (CDs). Auf der x-Achse stehen die sieben Rang-Bins und auf der y-Achse der Anteil der Merkmale, die in dem jeweiligen Bin liegen.	51

Algorithmenverzeichnis

2.1	FKNN, nach [8]	20
-----	--------------------------	----

Literaturverzeichnis

- [1] BAI, J., K. LUO, J. PENG, J. SHI, Y. WU, L. FENG, J. LI und Y. WANG: *Music emotions recognition by cognitive classification methodologies*. In: *2017 IEEE 16th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, S. 121–129, July 2017.
- [2] BILMES, J. A.: *Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm*. Doktorarbeit, Massachusetts Institute of Technology, 1993.
- [3] DE HAAS, W. B., R. C. VELTKAMP und F. WIERING: *Tonal Pitch Step Distance: a Similarity Measure for Chord Progressions..* In: *ISMIR*, S. 51–56, 2008.
- [4] DEFFERRARD, M., K. BENZI, P. VANDERGHEYNST und X. BRESSON: *Fma: a dataset for music analysis*. arXiv preprint arXiv:1612.01840, 2016.
- [5] FERNÁNDEZ, F. und F. CHÁVEZ: *Fuzzy Rule Based System Ensemble for Music Genre Classification*. In: MACHADO, P., J. ROMERO und A. CARBALLAL (Hrsg.): *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, S. 84–95, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [6] GUYON, I., S. GUNN, M. NIKRAVESH und L. ZADEH (Hrsg.): *Feature Extraction, Foundations and Applications*. Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, 2006.
- [7] JERRI, A. J.: *The Shannon sampling theorem—Its various extensions and applications: A tutorial review*. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.
- [8] KELLER, J. M., M. R. GRAY und J. A. GIVENS: *A fuzzy K-nearest neighbor algorithm*. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(4):580–585, July 1985.
- [9] KOHAVI, R.: *A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection*. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95*, S. 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

- [10] LARTILLOT, O.: *MIRtoolbox 1.4 User's Manual*. Finnish Centre of Excellence in Intedisciplinary Music Research and Swiss Center for Affective Sciences, 2012.
- [11] MAUCH, M. und S. DIXON: *Approximate Note Transcription for the Improved Identification of Difficult Chords*. In: *ISMIR*, 2010.
- [12] MAUCH, M. und M. LEVY: *Structural change on multiple time scales as a correlate of musical complexity*. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011*, S. 489–494, 2011.
- [13] MCKAY, C.: *Automatic Music Classification with jMIR*. Doktorarbeit, McGill University, 2010.
- [14] ROSSING, T. D., P. WHEELER und R. MOORE: *The Science of Sound*. Addison Wesley, dritte Aufl., 2002.
- [15] THEIMER, W., I. VATOLKIN und A. ERONEN: *Definitions of audio features for music content description*. Tech. Univ., Fak. für Informatik, 2008.
- [16] VATOLKIN, I.: *Improving Supervised Music Classification by Means of Multi-Objective Evolutionary Feature Selection*. Doktorarbeit, Technische Universität Dortmund, Dortmund, 2013.
- [17] VATOLKIN, I., G. BONNIN und D. JANNACH: *Comparing audio features and playlist statistics for music classification*. In: *Analysis of Large and Complex Data*, S. 437–447. Springer, 2016.
- [18] VATOLKIN, I. und G. RUDOLPH: *Interpretable Music Categorisation Based on Fuzzy Rules and High-Level Audio Features*. In: LAUSEN, B., S. KROLAK-SCHWERDT und M. BÖHMER (Hrsg.): *Data Science, Learning by Latent Structures, and Knowledge Discovery*, S. 423–432, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [19] VATOLKIN, I., W. THEIMER und M. BOTTECK: *Partition based feature processing for improved music classification*. In: *Challenges at the interface of data analysis, computer science, and optimization*, S. 411–419. Springer, 2012.
- [20] VATOLKIN, I., W. THEIMER und G. RUDOLPH: *Design and comparison of different evolution strategies for feature selection and consolidation in music classification*. In: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, S. 174–181. IEEE, 2009.
- [21] VATOLKIN, I., W. M. THEIMER und M. BOTTECK: *AMUSE (Advanced MUSic Explorer)-A Multitool Framework for Music Data Analysis..* In: *ISMIR*, S. 33–38, 2010.

- [22] WEIHS, C., D. JANNACH, I. VATOLKIN und G. RUDOLPH: *Music Data Analysis: Foundations and Applications*. Chapman & Hall/CRC, erste Aufl., 2016.
- [23] YANG, Y.-H., C.-C. LIU und H. H. CHEN: *Music Emotion Classification: A Fuzzy Approach*. In: *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, S. 81–84, New York, NY, USA, 2006. ACM.
- [24] ZADEH, L. A.: *Fuzzy sets*. Information and control, 8(3):338–353, 1965.
- [25] ZIEGENRÜCKER, W.: *ABC Musik: allgemeine Musiklehre; 446 Lehr-und Lernsätze*. Breitkopf & Härtel, 2009.

Eidesstattliche Versicherung

Ginsel, Philipp

Name, Vorname

183866

Matr.-nr.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Bachelorarbeit mit dem Titel

**Strukturelle Komplexitätsmerkmale zur Musikgenre-Klassifikation mit
Fuzzy- k -nearest-Neighbors**

selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, den 1. Oktober 2018

Ort, Datum

Unterschrift

Belehrung:

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/ die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -)

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird gfs. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin“) zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Dortmund, den 1. Oktober 2018

Ort, Datum

Unterschrift

