Skalierbare automatische Klassifikation von Musikstücken mit individuellen Genre-Modellen

Rolf Engelhard[1], Manuel Möller[2], Stephan Baumann[3]

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), 67663 Kaiserslautern, Deutschland

[1] *E-Mail: engelhard@dfki.de*, [2] *E-Mail: manuel.moeller@dfki.de*, [3] *E-Mail: stephan.baumann@dfki.de*

Einleitung

Musikalische Genres sind Kategorien, die weder standardisiert, noch auf bestimmte, allgemeingültige Unterscheidungsmerkmale zurückzuführen sind. Sie werden nach subjektiven Gesichtspunkten vergeben [1]. Das hier vorgestellte System *jGenre* trägt diesem Rechnung, indem es das Genreverständnis des Benutzers anhand einer bereits klassifizierten Menge von Musikstücken modelliert und auf noch nicht klassifizierte Musikstücke anwendet.

Obwohl sich mittlerweile viele Ansätze zur automatischen Genreklassifikation in der Literatur finden, gibt es hierzu bisher nur sehr wenige Softwaresysteme, die frei zugänglich, benutzerfreundlich und skalierbar sind. Das hier vorgestellte Softwaresystem jGenre soll diese Lücke schließen. Unsere Implementierung ist Plattformunabhängig, auf Multiprozessorarchitekturen optimiert und unterstützt derzeit die Audioformate WAV, AIFF, MP3, FLAC und VORBIS. Lizensiert ist es unter der GNU General Public License.

Systemarchitektur

Die Genreklassifikation basiert auf dem weithin benutzten Verfahren, Deskriptoren aus den Audiodaten zu berechnen und einer *Support Vector Machine* zuzuführen. Dazu werden die Audiodateien in ein einheitliches Format (WAV) überführt (vergleiche Abb. 1). Motiviert ist dies durch eine einfache Erweiterung des jGenres auf neue Audioformate, der Reduzierung von Berechnungszeit (Downsampling der Abtastfrequenz und Downmixing zu Mono) sowie der Verringerung des benötigten Arbeitsspeichers.

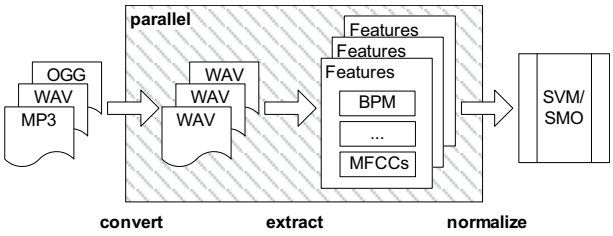

Abbildung 1: Prinzipieller Ablauf des jGenres. In der Konvertierungsphase werden alle Musikdateien in ein einheitliches Format dekodiert, in der Extraktionsphase die Features berechnet. Diese Schritte erfolgen für unterschiedliche Musikdateien parallel. Nach der abschließenden Normalisierungsphase werden die Featurevektoren der Support Vector Machine zugeführt.

Nebenläufigkeit

Besonderer Augenmerk wurde bei der Implementierung auf Skalierbarkeit, durch Aufteilung der Berechnung in mehrere nebenläufige Threads, gelegt. Multithreading findet sich bei der Dekodierung der Eingabedaten in das WAV-Format, die parallel zur Extraktion der Features bei bis zu 16 – schon dekodierter – Musikdateien stattfindet. Die Berechnung der Features erfolgt in kleinen einzulesenden Datenblöcken und verringert den benötigten Arbeitsspeicher signifikant. Zur Vermeidung einer Überführung der Extraktionsthreads in einen Wartezustand, wird während der aktiven Berechnung der nächste zu betrachtende Datenblock gecached.

Deskriptoren

Insgesamt wurden sechs Deskriptoren implementiert:

Beats Per Minute (BPM) – errechnet anhand des in [2] vorgestellten Verfahrens eine Liste der Zeitpunkte von Taktschlägen. Daraus leitet sich das Tempo des Musikstückes (in BPM) ab.

Length – liefert die logarithmisch skalierte Länge eines Musikstückes.

AudioPower [3] – beschreibt die subjektiv empfundene Lautheit eines Musikstückes. Eine hohe Dynamik des Musikstückes führt zu hohen Pegelsprüngen und einer hohen Standardabweichung.

AudioSpectrumCentroid und AudioSpectrumSpread [3] – liefern die Form und Verteilung des Spektrums. Es lässt sich unter anderem auf das Verhältnis hoher zu tiefer Frequenzen schließen.

Mel Frequency Cepstral Coefficients (MFCCs) – beschreiben die inverse Fouriertransformierte des melskalierten Spektrums [4]. Das Ergebnis ist (analog zu [5]) ein Vektor, bestehend aus den Mittelwerten der ersten n Koeffizienten und der entsprechenden Kovarianzmatrix Cov. Aufgrund der Symmetrie von Cov werden nur die Werte $Cov_{ij}, i \leq j$ beachtet.

Lernen

Vor Anwendung des maschinellen Lernverfahrens, werden die MFCC-Werte über alle Musiktitel standardisiert (zero-mean-unit-variance). Dies führt unter der Verwendung der euklidischen Distanz im Kernel der Support Vector Machine zum Gebrauch der Mahalanobis-Distanz [5].

Das eingesetzte maschinelle Lernverfahren ist eine Spezialform der Support Vector Machine: genutzt wird der *Sequential Minimal Optimization (SMO)* Algorithmus des Rapidminer-Frameworks [6]. Das Multiklassenproblem wird darin durch Bildung eines binären Entscheidungsbaums gelöst.

Ergebnisse

Zur Ermittlung der Klassifikationsgüte wurden drei exemplarische Testmengen zusammengestellt und jeweils durch *10-Fold Cross Validation* getestet (siehe Tabelle 1).

baumann200 202 Musiktitel, händisch in vier Genres („Misc“ (40%), „Jazz“(26%), „Rock“ (26%), „New Age“ (8%)) eingeteilt. Die Klassifikationsgüte $69,26\% \pm 9,75\%$ erklärt sich durch das Genre „Misc“, welches einerseits den größten Anteil und andererseits eine sehr heterogene Menge von Musikstücken darstellt. Der SVM war es damit nicht möglich, die benötigte Generalisierung durchzuführen.

engelhard313 besteht aus 313 Musiktitel, die per Interpret in vier Genres eingeteilt wurden („Techno“ (27%), „Rap“ (26%), „Rock“ (23%), „Soundtrack“ (23%)). Ziel dieser Zusammenstellung war zu testen, wie akkurat das System bei einer annähernden Gleichverteilung der Titel mit gleichzeitig sich stark unterscheidenden Genres klassifiziere. Die überaus hohen Ergebnisse von $97,70\% \pm 2,56\%$ entsprechen den damit verbundenen Erwartungen.

uspop2002 (subset) [7] besteht aus 895 Musiktitel (Teilmenge von uspop2002), die per Interpret/Album-Tupel in fünf Genres eingeteilt wurden („Rock“ (69%), „Rap“ (10%), „Electronica“ (10%), „R&B“ (9%), „New Age“(2%)). Auffallend ist das Übergewicht der „Rock“-Titel. Der Mittelwert der Ergebnisse liegt bei $84,02\% \pm 3,48\%$.

Damit klassifiziert jGenre auf dem State-Of-The-Art-Niveau der Music Information Retrieval-Forschung.

Tabelle 1: Klassifikationsgüte des jGenre bei Anwendung von 10-Fold Cross Validation auf jeweils drei Testmengen. Die Werte entsprechen dem State-Of-The-Art der Genreklassifikation.

Testmenge	Deskriptormenge	
	MFCCs	alle
baumann200	67,31%	69,26%
engelhard313	95,73%	97,70%
uspop2002 (subset)	82,79%	84,02%
Ø	81,94%	83,66%

Die Implementierung der Konvertierungsphase der verschiedenen Dateiformate in das WAV-Format macht es einfach, jGenre für neue Formate zu erweitern. Bisher werden die Formate WAV, AIFF, MP3, FLAC und VORBIS sowie die Playlisten-Formate M3U und PLS unterstützt – eine, nach unserem Wissen, bisher einmalige Menge eines JAVA-basierten Softwaresystems der Genreklassifikation.

Die Auswirkungen der Multithreading-Fähigkeit wurden auf einem Dualcore-System getestet (max. RAM: 1024, max. temporärer Festplattenplatz: 750MB, 50 Titel, durchschnittliche Titeldauer 3:74 Min.). Wir konnten beim Schritt von einem auf mehrere Extraktionsthreads einen Performancegewinn von bis zu 54% messen. Insbesondere Läufe ohne Downsampling in der Dekodierungsphase (Samplerate = 44100 kHz) profitieren von der Parallelität. Pro Titel wurde bei 11025 kHz durchschnittlich 8 Sekunden, bei 22050 kHz 11 Sekunden und bei 44100 kHz 19 Sekunden Berechnungszeit benötigt.

Ausblick

Hauptaugenmerk bei der Weiterentwicklung werden Performance und Skalierbarkeit sein. So bietet die Einbindung der genutzten Drittsoftware und die Kommunikation zwischen Dekodierungs- und den Extraktionsthreads noch Optimierungspotential.

Literatur

[1] Craft, A. and Wiggins, G. and Crawford, T.: How many beans make five? The consensus problem in music-genre classification and a new evaluation method for single-genre categorisation systems. Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)

[2] Dixon, S.: Mirex 2006 Audio Beat Tracking Evaluation: Beatroot. Proceedings of the 2nd Music Information Retrieval Evaluation eXchange (MIREX'06)

[3] ISO/IEC 15938:2001(E) Information Technology — Multimedia Content Description Interface — Part 4: Audio. ISO, International Organization for Standardization.

[4] Pohle, T.: Extraction of Audio Descriptors and Their Evaluation in Music Classification Tasks. Master's thesis, University of Kaiserslautern, (2005)

[5] Mandel, M. and Ellis, D.: Song-Level Features and Support Vector Machines for Music Classification. Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)

[6] Mierswa, I. and Wurst, M. and Klinkenberg, R. and Scholz, M. and Euler, T.: YALE (now: RapidMiner): Rapid Prototyping for Complex Data Mining Tasks. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)

[7] Ellis, D. and Berenzweig, A. and Whitman, B.: The „uspop2002“ pop music data set. URL: http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html