

Musik-Empfehlungssysteme - ein Überblick

Thorben Voss

Lehrstuhl für Complex and Intelligent Systems, Universität Passau

vossthor@fim.uni-passau.de, maximilian.schmitt@uni-passau.de

Abstract

Diese Arbeit ermöglicht eine Übersicht verschiedener Musik-Empfehlungssysteme mit Blick auf moderne Streaming-Dienste. Die bekanntesten Herangehensweisen sind dabei das *kollaborative*- und das *inhaltsbasierte Filtern* [1]. Im Folgenden werden diese Methoden miteinander verglichen und anhand konkreter Ansätze erläutert. Zudem wird auf die Brauchbarkeit und Problematik der Herangehensweisen eingegangen.

Index Terms: Empfehlungssysteme, kollaboratives Filtern, inhaltsbasiertes Filtern, Musik-Ähnlichkeitsanalyse, i-Vektoren

1. Einführung

Wer heute neue Musik entdecken möchte und eine passable Internetverbindung besitzt, ist schon lange nicht mehr auf die Empfehlungen seiner Bekannten und deren Plattenregal angewiesen. Online werben viele Streamingplattformen mit ständiger Verfügbarkeit, einem gewaltigen Volumen an neuer Musik und insbesondere Methoden, neue und für den Benutzer interessante, Musiktitel zu finden.

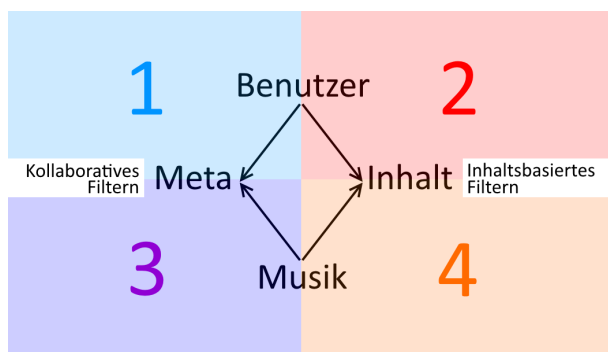


Abbildung 1: Filtermethoden und Informationen [2]

Die für die Musikempfehlung relevanten Informationen lassen sich für die spätere Verarbeitung in vier Bereiche unterteilen, wie es in Abbildung 1 zu sehen ist. **Feld 1** beschreibt alle Meta-Informationen, die über den Benutzer bekannt sind. Das können z.B. Alter, Geschlecht, aber auch Abspielverhalten oder Bewertungen sein, welche insbesondere beim kollaborativem Filtern eine tragende Rolle spielen. **Feld 2** enthält Informationen, warum dem Nutzer ein Musiktitel gefällt und wie es um seinen Geschmack steht. **Feld 3** beschäftigt sich mit Musik-Metadaten, wie etwa dem Künstler, dem Erscheinungsjahr oder der Titellänge, sowie Informationen, ob Musiktitel gerne zusammen abgespielt werden (siehe Abschnitt 2.3). **Feld 4** verwendet das Audiosignal um Komponenten, wie zum Beispiel verwendete Instrumente, den Takt oder die Varianz der Tonhöhe festzustellen. Die Analyse dieser Inhalte und Extraktion von Komponenten wird als inhaltsbasiertes Filtern bezeichnet.

2. Kollaboratives Filtern

Hier wird davon ausgegangen, dass sehr ähnliche Nutzer auch den gleichen Geschmack im Bezug auf die Wahl ihrer Objekte haben. Die Ähnlichkeit kann beispielsweise über Bewertungen der Nutzer festgestellt werden. Existieren nun zwei Nutzer, von denen der Erste zehn Musiktitel und der Zweite neun ähnlich bewertet hat, so wird die Bewertung des zweiten Nutzers für den zehnten Titel fast gleich ausfallen wie die des Ersten. Ist die Prognose eine gute Bewertung könnte sie dem zweiten Nutzer als Empfehlung angezeigt werden.

Der Vorteil hiervon ist, dass die Inhalte der zu bewertenden Objekte vernachlässigt werden können — in unserem Fall das Audiosignal des Musiktitels. So lässt sich dieses Empfehlungssystem auch in anderen Bereichen anwenden [3].

Direkte Bewertungen Direkte Bewertungen gehen vom Nutzer bewusst aus und werden meist dauerhaft in einem Nutzerprofil hinterlegt, weshalb eine Anmeldung im System erforderlich ist. Recht beliebte Bewertungssysteme wären die Angabe von Sternchen (z.B. x von maximal 5), exaktere Angaben, wie etwa eine prozentuale oder ein einfaches 'Gefällt mir' bzw. ein Daumen nach oben (Spotify).

Indirekte Bewertungen Speziell für die Musik wäre hier das Abspielverhalten zu erwähnen. Wird ein Titel vom gleichen Nutzerprofil mehrmals abgespielt oder gar die Lautstärke erhöht, so gefällt er ihm vermutlich. Wird der Titel nach kurzer Zeit übersprungen, pausiert oder abgebrochen ist er nicht besonders daran interessiert.

2.1. Berechnungsansätze

Nun gibt es zwei Herangehensweisen diese Bewertungsinformationen zu verarbeiten: einen **speicherbasierten**- und einen **modellbasierten** Ansatz. Letzterer wird verwendet, wenn das Volumen an Bewertungsdaten einen Zugriff in akzeptabler Zeit über eine Matrix nicht mehr zulässt. Man versucht dann die bisherigen Bewertungen auf ein vorhersagendes Modell zu übertragen, wie etwa *clustering* in Kombination mit dem *Naive Bayes Modell* oder *latent factor models* um Objekt- und Nutzer-spezifische Eigenschaften zu charakterisieren.

Der speicherbasierte Ansatz beruht auf heuristischen Algorithmen [4], die Vorhersagen über Bewertungen aufgrund von abgegebenen Bewertungen der N ähnlichsten Nutzer oder Objekte treffen. Diese werden in eine Nutzer-Objekt Matrix eingetragen. Eine Zeile entspricht einem Nutzer, die in den Spalten eingetragenen Zahlen den abgegebenen Bewertungen. Man unterscheidet zwischen **nutzerbasierten** und **objektbasierten** Ansätzen, die sich auf die im Algorithmus eingebaute Ähnlichkeitsfunktion beziehen. In diesem Kontext möchte ich genauer auf nutzerbasierte Herangehensweisen eingehen, da die Bestimmung der Ähnlichkeit zwischen Objekten im inhaltsbasierten Filtern auch ohne zusätzliche Meta-Daten funktioniert.

2.2. Nutzerbasiertes kollaboratives Filtern

Die gesuchte Bewertung (rating) $r_{u,o}$ für den Benutzer (user) u für das Objekt o wird über die Aggregationsfunktion $aggr(r_{u',o})$ berechnet. \hat{U} sei im Folgenden die Menge der N ähnlichsten Nutzer, die bereits eine Bewertung zu dem Objekt o abgegeben haben (siehe Formel 1).

$$r_{u,o} = \text{aggr}_{u' \in \hat{U}}(r_{u',o}) \quad (1)$$

Die genaue Form der Aggregationsfunktion wird durch Normalisierungen und Gewichtung der Ähnlichkeit bestimmt [1][4]. Im Folgenden werden schrittweise Komponenten hinzugezogen. Die Gleichung (2) berechnet trivialerweise den Durchschnitt der Bewertungen von N -vielen Nutzern.

$$r_{u,o} = \frac{1}{N} \sum_{u' \in \hat{U}} r_{u',o} \quad (2)$$

$$r_{u,o} = k \sum_{u' \in \hat{U}} \text{sim}(u, u') * r_{u',o} \quad (3)$$

$$r_{u,o} = \bar{r}_u + k \sum_{u' \in \hat{U}} \text{sim}(u, u') * (r_{u',o} - \bar{r}_{u'}) \quad (4)$$

Die Gleichungen (3) und (4) gewichten mit $\text{sim}(u, u')$ den Grad der Ähnlichkeit zwischen den Benutzern u und u' . Die Ähnlichkeitsfunktion wird anschließend angesprochen. k dient als normalisierender Koeffizient und wird als $\frac{1}{\sum_{u' \in \hat{U}} |\text{sim}(u, u')|}$ definiert. \bar{r}_u bzw. $\bar{r}_{u'}$ bezeichnen die durchschnittliche Bewertung des Benutzers u bzw. u' . $\bar{r}_u = \frac{1}{O_u} \sum_{o \in O_u} r_{u,o}$.

Wie schnell deutlich werden sollte berücksichtigt die *Formel 2* nicht den Geschmack einzelner Nutzergruppen, sofern die N -Nutzer beliebig gewählt wurden. Es ist aber möglich eine allgemeine Einschätzung zu einem Titel geben zu können, was man auch als 'Mainstreambewertung' bezeichnen könnte. Sind kaum Nutzer vorhanden ist diese Methode für den Anfang keine schlechte Wahl. *Formel 3* hingegen gewichtet Nutzer: desto höher der Grad der Ähnlichkeitsfunktion $\text{sim}(u, u')$, desto relevanter ist die Bewertung. k normalisiert das Ergebnis. Allerdings wird nicht beachtet, dass es pessimistische und gutmütige Bewerter gibt, welche durchschnittlich wenige, bzw. mehr 'Sterne' verteilen. Diese Problematik wird in *Formel 4* durch Einbeziehung der Abweichung der Bewertung $r_{u',o}$ zu der durchschnittlichen Bewertung $\bar{r}_{u'}$ des Nutzers u' behoben.

Die Form der *Ähnlichkeitsfunktion* $\text{sim}(u, u')$ kann unterschiedlich ausfallen. Häufig werden der korrelationsbasierte (5) oder kosinusbasierte Ansatz (6) angewandt [4][1].

$$\text{sim}(x, y) = \frac{\sum_{o \in O_{xy}} (r_{x,o} - \bar{r}_x)(r_{y,o} - \bar{r}_y)}{\sqrt{\sum_{o \in O_{xy}} (r_{x,o} - \bar{r}_x)^2 * \sum_{o \in O_{xy}} (r_{y,o} - \bar{r}_y)^2}} \quad (5)$$

$$\text{sim}(x, y) = \cos(x, y) = \frac{\sum_{o \in O_{xy}} r_{x,o} r_{y,o}}{\sqrt{\sum_{o \in O_x} r_{x,o}^2} \sqrt{\sum_{o \in O_y} r_{y,o}^2}} \quad (6)$$

O_{xy} ist die Menge an Objekten, die sowohl von Benutzer x , als auch Benutzer y bewertet wurden. In (5), dem korrelationsbasierten Ansatz wird der *Pearson-correlation-coefficient* zur Ähnlichkeitsberechnung verwendet. Der kosinusbasierte Ansatz (6) spannt für die Benutzer Vektoren mit Dimension der Anzahl von gemeinsamen Objekten auf. Der Kosinus des Winkels zwischen den Vektoren bestimmt deren Ähnlichkeit. Die

Berechnung der Ähnlichkeiten bei zunehmenden Nutzerzahlen wird aufgrund des Rechenaufwandes im Vorfeld erledigt oder die zu betrachtenden N -Nutzer mittels weiterer Meta-Daten z.B. nach Bundesland gefiltert um weniger Berechnungen durchführen zu müssen.

2.3. Objektbasiertes kollaboratives Filtern

In [1] werden empirische Studien nach [5] erwähnt, die Ähnlichkeiten der Objekte statt der Benutzer berechnen und performanter scheinen. Dabei soll die Ähnlichkeit von zwei Objekten hoch sein, wenn viele Nutzer existieren, die beide angesehen bzw. gehört haben. Hört ein neuer Nutzer nun nur eines, so werden ihm sofort die ähnlichen Objekte vorgeschlagen, ohne die Ähnlichkeit über die vorgestellte Matrix zu allen Nutzern und allen Objekten bestimmen zu müssen. Die Objektähnlichkeit kann aber auch ohne Zusatzinformationen direkt über die Analyse des Audiosignals bestimmt werden, weshalb hier nicht weiter darauf eingegangen wird.

2.4. Problematik bei kollaborativen Filtermethoden

kalter Start Sind keine Meta-Daten vorhanden kann nicht kollaborativ gefiltert werden. Das ist immer dann der Fall, wenn das System neu aufgesetzt wird. Die ersten Nutzer können daher noch nicht von dem Empfehlungssystem profitieren. Eine Lösung wäre das Trainieren mit Beispieldaten oder die Verwendung 'größerer' Aggregationsfunktionen (Formel 2), sowie Einbeziehung von Sensordaten, wie Zeit oder Ort um erste Empfehlungen geben zu können [1].

Sparsamkeit Hier gibt es viele Benutzer, aber trotzdem wenig Bewertungen. Ist man als Anbieter zu aufdringlich mit Bewertungsaufforderungen und verwendet ausschließlich direkte Bewertungen, so kann es passieren, dass einfach zu wenige vorhanden sind, um eine aussagekräftige Empfehlung mittels kollaborativem Filtern abgeben zu können. Eine Lösung wäre die Verwendung indirekter Bewertungen (siehe Seite 1). Nach [6] geben zufriedene Nutzer sogar aus Dankbarkeit Bewertungen ab, um dem System etwas gutes zu tun.

Schneeballeffekt Ein Titel, der bereits bewertet wurde, wird wahrscheinlicher angezeigt, als einer mit wenigen Bewertungen. So wird ungewollt der bereits bewertete gefördert, indem er mehr Chancen bekommt bewertet zu werden. Eine Lösung wäre die Anzahl an Bewertungen mit einzubeziehen, sodass z.B. bei Gleichstand der Bewertung dem Objekt mit weniger Bewertungen die Chance gelassen wird.

Neue Musik Ein neuer Musiktitel wird dem System hinzugefügt, lässt sich aber aufgrund mangelnder Bewertungen noch keiner Nutzergruppe / keinem Geschmack zuordnen. Eine Lösung wäre für neue Titel eine eigene Sparte zu eröffnen um Bewertungen zu generieren. Zudem lässt sich über das objektbasierte kollaborative Filtern auch abschätzen, ob der Künstler zu einer Nutzergruppe passt. Mochte die Gruppe die letzten drei Alben eines Künstlers, ist sie voraussichtlich auch an dem neuen Album interessiert.

Neue und außergewöhnliche Nutzer Ein Nutzer kommt neu hinzu oder hat so einen ausgefallenen Geschmack, dass er sich keiner Nutzergruppe über die Ähnlichkeitsfunktion $\text{sim}(x, y)$ zuordnen lässt bzw. nur geringe Ähnlichkeit aufweist. Ein Ansatz wäre die Nutzerebene zu verlassen und mehr auf objektbasiertes Filtern zu bauen (kollaborativ und inhaltsbasiert).

3. Der Benutzerkontext

Neben den Bewertungen und dem Nutzerverhalten liegen meist weitere Informationen wie Sensordaten vor, die es sich lohnt für eine gute Empfehlung miteinzubeziehen.

Ortsangaben helfen besonders für landesweit und kontinental häufig abgespielte Mainstream Titel [7]. Bei der Musik-Streaming Plattform Spotify können so die Top 50 und Viral 50 der Welt dem Land zugeordnet werden um landesspezifische Charts über das Abspielverhalten zu generieren.

Zeit hat einen massiven Einfluss auf das Hörverhalten des Nutzers. So möchte man am Morgen eher lockere und leicht eingängige Musik, während Freitagabends basslastige Musik und nachts Schlafmusik gewünscht werden [6]. So kann konkreter jedes Nutzerprofil in weitere zeitabhängige Unterprofile unterteilt werden um dem Kontext gerecht zu werden. Man spricht hier vom *micro-profiling* [8].

Alter und Geschlecht Nach [9] hören jüngere Nutzer, im Gegensatz zu 41-jährigen und älteren Hörern, lieber Charts, populäre und moderne Musik. Daher sollte es für sie eine Sparte 'Mainstream' geben, für die älteren Nutzer nicht unbedingt. Außerdem ließen sich popularitätsbasierte Empfehlungen besser bei Frauen, als bei Männern anwenden.

4. Inhaltsbasiertes Filtern

Im Bezug auf die Musik bei Musikempfehlungssystemen wird hier versucht Komponenten aus dem Audiosignal des Musiktitels zu extrahieren (Seite 1, Abbildung 1, Feld 4) und diese mit dem Geschmack (Feld 2) des potentiellen Hörers abzugleichen. Ein Nutzer, der z.B. schon viel tiefe basslastige Musik im 4/4-Takt gehört hat wird dies auch weiterhin gerne tun. Die Berechnung erfolgt über eine Brauchbarkeitsfunktion des Benutzers u für das Objekt o über eine Menge an Attributen. Entsprechen die Werte der Attribute den Vorlieben des Nutzers wird der Titel empfohlen [1]. Zudem ist es möglich die Ähnlichkeit der Attribute zwischen zwei Musiktiteln festzustellen. Spotify setzt letzteres in einer eigenen Sparte um [6]: [Browse] - [Entdecken] - "Weil du X gehört hast" [empfehlen wir dir folgende Titel - Anm. d. Verf.], Stand Juli 2016. X ist ein bereits gehörter Titel.

4.1. Gaussian mixture model

Als grundlegendes 'Universal Background Model' (UBM) für inhaltsbasierte Filtertechniken dient meist das Gaussian mixture model [10] (GMM), welches eine akkustische Modellierung von Musikstücken ermöglicht. Bei einem GMM handelt es sich um ein Wahrscheinlichkeitsmodell mit der Annahme, dass sich alle Datenpunkte innerhalb einer endlichen Anzahl an gaußscher Verteilungen befinden [11]. Das GMM wird mittels einer Datenbank an Musiktiteln trainiert und für jeden Titel wird ein sogenannter GMM-Supervektor mit fester Länge generiert [10]. Das Verfahren dieser Generierung wird in [12] vorgestellt.

Das Problem bei der Verwendung von GMM-Supervektoren ist, dass bei zunehmender Anzahl an gaußschen Verteilungen, die für die Abgrenzung der Genres bei der Genreklassifizierung dienen, die Dimension des GMM-Supervektors drastisch zunimmt. Zur Lösung dieses Problems können Verfahren zur Dimensionsreduktion, wie PCA (Principal Component Analysis) angewandt werden, die aber oft mit Informationsverlust einhergehen und nach [10] nicht besonders effektiv zu sein scheinen. Ein weiterer Nachteil der GMM-Supervektoren ist, dass nicht zwischen den Variabilitäten unterschieden werden kann und somit immer

jeglicher Unterschied berücksichtigt wird. Um eine Trennung von gewünschten Faktoren (z.B. Genre-Variabilität) und ungewünschten (z.B. Song-Variabilität) vorzunehmen werden daher Methoden der Faktoren Analyse (FA) angewandt.

4.2. Methoden der Faktorenanalyse

Für die Faktorenanalyse wird ein Modell verwendet, dass man auch als GMM-Supervektorraum bezeichnen könnte [10]. Die Faktoren des Genres und des Musiktitels werden so definiert, dass zwei Musiktitel desselben Genres auch die gleichen Werte der genrespezifischen Faktoren aufweisen. Ein erster Ansatz sieht so aus: $M(g) = m + O_g$. Dabei sei $M(g)$ ein vom Genre abhängiger GMM-Supervektor, m ein vom Genre unabhängiger Vektor der anschließend trainiert wird, g das Genre und O_g ein unbekannter, nicht berechenbarer oder trainierbarer Ausgleichsvektor [10]. Das eben beschriebene Verfahren der Extraktion von genreabhängigen GMM-Supervektoren heißt **Maximum a posteriori (MAP)** und nimmt an, dass es für jedes Genre g nur einen Supervektor $M(g)$ gibt. Zu erwähnen wäre noch das Verfahren *eigenvoice MAP*, welches unter gewissen Annahmen ein schnelleres Trainieren als MAP ermöglicht. Eine Trennung von Song- und Genrefaktoren wurde erst durch die **Joint Factor Analysis (JFA)** möglich, welche zusätzliche Unterräume für die Variabilitäten aufspannt, aber immer noch nicht sauber trennen kann (Song-Faktoren enthalten Informationen über Genres). Das für uns interessante Faktorenanalysemodell heißt **Total Variability Space (TVS)** [13] und die darin enthaltenen Vektoren werden als **i-Vektoren** bezeichnet.

4.3. i-Vektoren

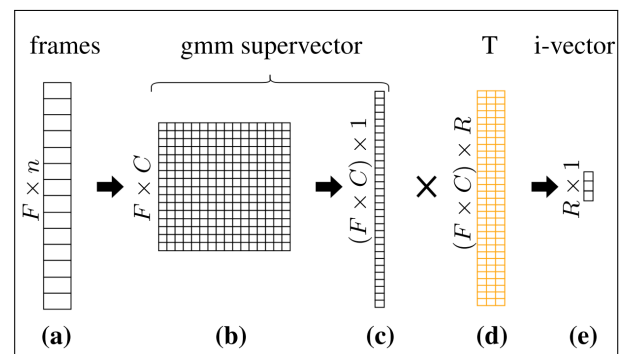


Abbildung 2: Graphische Repräsentation der Vektoren während der i-Vektoren Faktorenanalyse. Entnommen aus [10].

Die *frames* in (a) sind Ausschnitte des zu analysierenden Musikstücks. F ist die Anzahl akkustischer Komponenten, C die der gaußschen Verteilung. R ist der Rang der TVS-Matrix und letztendlich die Dimension des i-Vektors. Die aus (a) F -vielen extrahierten Komponenten werden mit einem GMM des Ranges C zu GMM-Supervektoren (b) der Dimension $F \times C$ verrechnet und in eine $(F \times C) \times 1$ Matrix umgewandelt. Die TVS-Matrix T mit Dimension $(F \times C) \times R$ (c) wurde zuvor auf eine gewisse Anzahl an Songs trainiert und ermöglicht nun die gewünschte Dimensionsreduktion des GMM-Supervektors auf R , welcher nun als i-Vektor bezeichnet wird. Nach [10] (2015) enthält dieser Vektor trotz deutlicher Dimensionsverminderung noch die gewünschten Eigenschaften und ermöglicht eine angemessene Ähnlichkeitsanalyse sowie die bisher effizienteste Künstlerklassifikation auf dem *artist20*-Datensatz.

4.4. Music Information Retrieval

Es wurde nun auf die *i*-Vektoren aufmerksam gemacht, welche aber nur eine, recht moderne, von vielen Herangehensweisen ist. Music Information Retrieval heißt die Forschungsdisziplin, die sich seit gut 18 Jahren mit der Extraktion von Informationen aus Musikstücken zur Weiterverarbeitung beschäftigt. Die Methoden sind dabei ausgesprochen vielfältig und bewegen sich in Forschungsgebieten der Signalverarbeitung, Kognitionswissenschaften und künstlichen Intelligenz bis hin zur Musiktheorie und Analyse [14]. Die Ansätze lassen sich, ähnlich wie beim kollaborativem Filtern, in speicherbasiert-heuristische und modellbasierte Methoden unterteilen. Beispiele für letztere wären der Bayes-Klassifikator, Clustering, tiefe neuronale Netze und Entscheidungsbäume. Auf der anderen Seite wäre Term Frequency-Inverse Document Frequency (TF-IDF) zu erwähnen [4]. Die Verwendung neuronaler Netze wurde schon früh theoretisch angedacht, ist aber erst mit heutiger Rechenleistung umsetzbar. Ein konkreter Ansatz wäre die Ähnlichkeitsanalyse mittels einer kovarianten eingeschränkten Boltzmann-Maschine, welche auf eine Musikdatenbank trainiert wird [15].

4.5. Problematik bei inhaltsbasierten Filtermethoden

Vielfalt Selbst wenn die inhaltsbasierten Filtermethoden sehr gute Ergebnisse liefern enthalten die empfohlenen Musikstücke im Optimalfall die gleichen Komponenten, wie bereits gehörte. Somit bewegt man sich oft im selben Genre mit seinen Lieblingskomponenten, anstelle etwas wirklich neues zu entdecken. Eine Lösung wäre einige Ausreißer miteinzubeziehen, die sich beispielsweise nur in einer Komponente, wie einem verwendeten Instrument von den bisherigen Musiktiteln abhebt.

Qualität Verwendet man rein inhaltsbasiertes Filtern kann ein Titel vorgeschlagen werden, der zwar von den betrachteten Komponenten sehr ähnlich ist, aber schrecklich klingt und dem Nutzer nicht gefallen wird. Dieses Problem stellt die Grundannahme des inhaltsbasierten Filterns infrage: ob ähnliche Komponenten überhaupt bedeuten, dass der Nutzer den Musiktitel hören möchte oder die für ihn wichtigen Komponenten überhaupt berücksichtigt wurden.

Neuer Nutzer Vor diesem Problem bleibt auch der inhaltsbasierte Ansatz nicht verschont. Wird ein neuer Nutzer dem System hinzugefügt kann keine Aussage getroffen werden, was er hören möchte. An dieser Stelle sollten ihm einige, möglichst verschiedene Titel präsentiert werden, um möglichst schnell an seine bevorzugten Komponenten zu kommen und ähnliche Titel zu empfehlen. Das Problem mit neuer Musik gibt es im Gegensatz zum kollaborativem Filtern nicht.

5. Hybridsysteme

Wie man an der Problematik des jeweiligen Filtertyps ablesen kann, ist es häufig sinnvoll beide (inhaltsbasierte und kollaborative) Filtermethoden anzuwenden. Das Zusammenspiel der Methoden bleibt eine Designentscheidung [16]. Beispiele:

Gewichtet Beide Methoden werden parallel ausgeführt und geben gewichtete Stimmen über zu empfehlende Titel ab. Die *N*-Titel mit den meisten Stimmen werden vorgeschlagen.

Umschaltend beginnt mit einer zufälligen Auswahl an viraler Musik, verwendet bei wenigen Bewertungen inhaltsbasiertes Filtern und wechselt ab einer bestimmten Gesamtähnlichkeit $\sum_{u' \in \hat{v}} |sim(u, u')|$ (Formel 4) auf das kollaborative Filtern.

Getrennt Sofern in der Benutzeroberfläche möglich, können die Ergebnisse einzelner Filtermethoden in eigenen Bereichen angezeigt werden. So könnte man oben einen gewichteten Ansatz, darunter einen inhaltsbasierten aufgrund der zuletzt gehörten darstellen und darunter anzeigen, was andere ähnliche Nutzer gerade hören.

Ergänzend Hier können die beiden Methoden gegenseitig voneinander profitieren. Das inhaltsbasierte Filtern erkennt Meta-Daten, wie den Künstler, die dann textuell als Meta-Daten gespeichert bzw. zu den bisherigen Informationen ergänzt werden und dem Meta-Filtern über Suchmasken dienen. Umgekehrt kann das inhaltsbasierte Filtern darauf verzichten ähnliche Objekte anzuzeigen, die von der Nutzergruppe, in der sich der Hörer befindet, nicht gemocht werden.

Kaskadiert Von 1000 Musiktiteln in der Datenbank empfiehlt die eine Methode 50%, von denen die zweite Methode wieder 50% auswählt und so weiter.

Kombiniert Die reinste Form Hybrider Empfehlungssysteme kombiniert die Informationen bereits im Algorithmus. So könnte die Bestimmung der Ähnlichkeit zweier Benutzer beim kollaborativem Filtern die Anzahl an akustischen Komponenten miteinbeziehen, die von beiden Nutzern präferiert werden.

Mit diesen Methoden ist es möglich alle Probleme, bis auf das des neuen Nutzers, zu relativieren. Daher ist zu empfehlen immer einen hybriden Ansatz zu implementieren.

6. Zusammenfassung und Ausblick

In dieser Arbeit wurden die bedeutendsten Ansätze für Musik-Empfehlungssysteme vorgestellt. Das *kollaborative Filtern* ermöglicht eine Empfehlung aufgrund von Metadaten, wie Bewertungen, ähnlicher Nutzer um eine heuristische Abschätzung über unseren Geschmack geben zu können. Gefällt es Nutzern, die sich ähnlich verhalten wie wir, mögen wir es vermutlich auch. Diese Methode ist leider von einer gewissen Anzahl an Bewertungen und Nutzern abhängig, was bei Neueinführung eines Systems kaum umsetzbar ist. Das *inhaltsbasierte Filtern* kann ohne Bewertungen akustische Komponenten aus dem Audiosignal filtern, um dann die Ähnlichkeit zweier Musikstücke festzustellen. Das Problem hierbei ist mehr, ob man intuitiv dazu verleitet ist ähnliche Musik zu hören, wenn man neue Musik empfohlen bekommen möchte. Zudem ist die Extraktion der Komponenten recht komplex, rechenaufwändig und muss oft für Audiosignale angepasst werden, während sich kollaborative Filtermethoden auf jegliche Objekte anwenden lassen. Um den meisten Problemen zu entgehen lassen sich hybride Ansätze verwenden, die beide Ansätze auf viele Arten verknüpfen können um dem Nutzer die bestmögliche Empfehlung zu ermöglichen. Neue Nutzer bleiben problematisch.

In Zukunft wird man wohl immer häufiger mit Empfehlungssystemen konfrontiert, da sich bald jeglicher Konsum auch online tätigen lässt und Streamingdienste andere Medien vertreiben. Es ist vom Nutzer abhängig, ob er einem Empfehlungssystem oder einer Person bei seiner Hörentscheidung vertraut, nur bleibt zu erwähnen, dass beide auch Eigeninteressen verfolgen können. Ein Empfehlungssystem sollte seine Algorithmen daher offen legen und auf Korruptierbarkeit überprüft werden. Es sollte erkennen, welche Bewertungen authentisch sind und welche neue Musik wirklich neu ist und in die Sparte darf. Daher sind in Zukunft Maßnahmen zur Qualitätssicherung der Systeme zu treffen um dem Benutzer mit progressiven Methoden auch weiterhin interessante Musik empfehlen zu können.

7. References

- [1] K. K. Gontarska, "Erweiterung klassischer Musikempfehlungssysteme um den Kontext des Wetters," Bachelorarbeit, Freie Universität Berlin, Deutschland, 2016.
- [2] M. S. Peter Knees, "Music retrieval and recommendation. a tutorial overview," 2015.
- [3] M. Abfalter, "Empfehlungssysteme für Audio-Communities am Beispiel des Projekts miii," Diplomarbeit, Fachhochschul-Masterstudiengang Digitale Medien in Hagenberg, 2010.
- [4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 737–738, 2005.
- [5] M. Deshpande and G. Karypis, "Item-Based Top-N Recommendation Algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 172–175, 2004.
- [6] wired.de, "Zukunft der musik / so funktionieren die musikvorschläge von spotify," <https://www.wired.de/collection/latest/wie-die-musik-vorschlagsmaschine-von-spotify-funktioniert>.
- [7] A. V. D. H. M. T. Katayoun Farrahi, Markus Schedl, "Impact of listening behavior on music recommendation," kollaboratives Filtern mit Einbindung von Kontinent und Land.
- [8] X. A. Linas Baltrunas, "Towards time-dependant recommendation based on implicit feedback," 2009, workshop on context-aware recommender systems (CARS'09).
- [9] M. Schedl, "Listener-aware music recommendation from sensor and social media data," 2015, joint European Conference on Machine Learning and Knowledge Discovery in Databases.
- [10] M. S. G. W. Hamid Eghbal-zadeh, Bernhard Lehner, "I-vectors for timbre-based music similarity and music artist classification," Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR), Málaga, Department of Computational Perception, Johannes Kepler University of Linz, Austria, 2015.
- [11] scikit learn, "Gaussian mixture models," <http://scikit-learn.org/stable/modules/mixture.html>.
- [12] D. T. Charbuillet, Christophe and G. Peeters, "Gmm-supervector for content based music similarity," 2011, international Conference on Digital Audio Effects, Paris, France.
- [13] e. a. DEHAK, Najim, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [14] Wikipedia, "Music information retrieval," https://de.wikipedia.org/wiki/Music_Information_Retrieval.
- [15] C. O. Jan Schlüter, "Music similarity estimation with the mean-covariance restricted boltzmann machine," *Proceedings of the Tenth International Conference on Machine Learning and Applications*, 2011.
- [16] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.