Recommender Systems - Homework 03

Klappert Lukas, Scheffknecht Daniel, Sum Stephan

1. DEMOGRAPHIC FILTERING

Wir haben insgesamt drei demografische Filter für unseren CF implementiert: Basierend auf dem Alter, dem Geschlecht, sowie der Herkunft.

Wie auch beim CF übergeben wir bei allen Implementierungen den aktuellen User, die User-Artist-Matrix, max_items_to_predict und nearest_users_to_consider. Zusätzlich übergeben wir hier users_extended, das die Daten von C1ku_users_extended.csv enthält, auf deren Basis wir filtern. Alle Implementierungen haben außerdem gemeinsam, dass am Ende des Filterns alle anderen Einträge in der UAM gleich 0 gesetzt werden. Hat der gegebene User kein Alter/Geschlecht/Ort angegeben, rufen wir den ursprünglichen CF unverändert auf.

Der erste Filter basiert auf dem Geschlecht. Da es schwierig ist, die Ähnlichkeit des Geschlechts der User zu berechnen, filtern wir alle User heraus, die exakt das gleiche Geschlecht angegeben haben. Wenn der User ein Geschlecht angegeben hat, werden die Playcounts aller User mit anderem/keinem angegebenen Geschlecht 0 gesetzt.

Der zweite Filter basiert auf dem Alter. Um garantieren zu können, dass nach dem Filtern mindestens nearest_users_to_consider User übrig bleiben, haben wir uns gegen ein fixes Limit beim Altersunterschied entschieden. Stattdessen berechnen wir für alle User den (absoluten) Altersunterschied zum gegebenen User. Vorher setzen wir jedoch das Alter aller Nutzer, die kein Alter angegeben haben, auf 999. Dadurch ist der Altersunterschied der User, die kein Alter angegeben haben, in jedem Fall sehr groß, sodass diese keine Rolle spielen. Außerdem berechnen wir mittels scidist.cosine() die Ähnlichkeit zum gegebenen User.

Anschließend nutzen wir numpy.lexsort(), um nach mehreren Werten sortieren zu können. Zunächst sortieren wir aufsteigend nach Altersunterschied, sodass User mit dem geringsten Altersunterschied ganz oben stehen. Da es in den meisten Fällen einige User mit dem gleichen Alter gibt, sortieren wir anschließend die User mit dem gleichen Alter absteigend nach Ähnlichkeit (siehe Tabelle 1).

User-ID	Altersunterschied	Ähnlichkeit
28	0	0
241	0	0.7
123	1	0.5
951	2	0.6

Table 1: Zusammenhang von Ähnlichkeit und Alter

Nach dieser Sortierung nehmen wir die ersten nearest_users_to_consider und setzen die Playcounts aller anderen User auf 0. Durch diese Vorgehensweise können wir garantieren, dass nearest_users_to_consider User vorgeschlagen werden können.

Der dritte und letzte Filter nutzt den Ort. Um auch hier zu garantieren, dass nearest_users_to_consider User genutzt werden können, haben wir uns dagegen entschieden, nur Nutzer aus dem gleichen Land zu verwenden. Stattdessen berechnen wir, ähnlich wie beim Alter, die Entfernung zum aktuellen User. Hat der gegebene User keinen Ort angegeben, rufen wir den ursprünglichen CF wieder unverändert auf. Für andere User, die keinen Ort angegeben haben, setzen wir die Distanz auf einen sehr hohen Wert, sodass diese - wie beim Alter - für die spätere Berechnung keine Rolle spielen.

Die Entfernung berechnen wir anhand der Koordinaten mittels geopy.great_circle(). Anschließend gehen wir genauso vor wie beim Alter: Die User werden mittels numpy.lexsort() aufsteigend nach Distanz und absteigend nach Ähnlichkeit sortiert, anschließend die ersten nearest_users_to_consider gefiltert und alle anderen 0 gesetzt.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. EVALUATION DER RS

2.1 Why is there a performance difference between your two baseline recommenders?

Wie erwartet ist die selbst implementierte Random Baseline, die die Interpreten eines zufälligen Nutzers nimmt, besser, als komplett zufällig irgendwelche Interpreten vorzuschlagen. Der Grund ist der, dass sehr viele Interpreten Teil des "Long Tails" sind; also nur von sehr wenigen Nutzern gehört werden. Wenn wir aber einen zufälligen Nutzer nehmen und dessen Interpreten vorschlagen, so werden viele dieser Interpreten eher im "Short Head" sein und somit wahrscheinlicher auch von dem Nutzer, dem wir die Vorschläge machen, gehört worden sein.

2.2 Can you outperform the stand-alone CF and CB recommenders using the hybrid approach? Why? Or why not?

In unserem Fall haben die schlechten Ergebnisse des Lyrics-CB auch die Performance-Werte unseres Borda-Rank-Hybrids negativ beeinflusst, da sowohl der CF als auch der CB gleichberechtigt Einfluss fanden (jeder hat gleich viele Artists empfohlen). Der Hybrid konnte daher den Standalone-CF nicht schlagen.

2.3 For your CB recommender, did you encounter any problems during data acquisition? How did you resolve them? To which extent do you think the data source influences the recommendation quality?

Bei der Datenakquise mussten wir zum einen klassische 404-Fehlerseiten abfangen, da nicht zu jedem Song der Songtext verfügbar war. Zum anderen gab es einige Instrumental-Songs für die es natürlich ebenfalls keine Songtexte gab. Werden diese Fehlerfälle nicht abgefangen, erhalten die betroffenen Dokumente zueinander eine Similarity von 1, was die Ergebnisse signifikant verfälschen würde.

2.4 Which of the two, CF and its extension DF, performs better? Speculate about the reasons.

Keine unserer DF-Varianten konnte den Standalone-CF schlagen. Ein individueller Ähnlichkeitsvergleich des Musikgeschmacks auf Basis der Playcounts ohne demografische Vorfilterung scheint hier der Königsweg zu sein. Den Grund für diese Beobachtung sehen wir in der geringeren Aussagekraft demografischer Faktoren im Vergleich zum Playcount. In Situationen in denen der Playcount-Datenbestand nur sehr spärlich gefüllt ist (Cold-Start) ist die Suche ähnlicher User jedoch relativ schwierig. Hier könnte man einen DF-Filter als Fallback einsetzen.

2.5 Among the three DF variants, which one performs best? Any idea why?

Unsere Messungen ergaben, dass der DF-Gender der stärkste unter den drei DF Varianten darstellt (vergleiche Abbildung 1). Es ist daher anzunehmen dass es eine nicht zu unterschätzende Ähnlichkeit im Musikgeschmack gleicher Geschlechter gibt. Der Einfluss des Geschlechts auf den Musikgeschmack kann unterschiedlich erklärt werden. Denkbar wären evolutionär-bedingte Erklärungsversuche oder gesellschaftliche bzw. kulturelle Prägungseinflüsse. Auch der Einfluss der Landes hat einen messbaren Einfluss auf den Musikgeschmack der User, wie die relativ guten Performance-Werte des DF-Country zeigen. Das Alter hat dagegen einen deutlich geringeren Einfluss gezeigt. Es ist daher anzunehmen, dass ein Großteil der Artists über alle Altersgrenzen hinweg gehört werden.

Jedoch ist trotz alledem zu beachten, dass CF ohne Demographic Filtering immer noch am besten abschneidet. Am wichtigsten ist es in unserem Fall, User anhand ihres Musik-Geschmackes zu vergleichen, und Demographic Filtering führt nur zu einer Verschlechterung des Ergebnisses. Dies würde auch erklären, wieso Gender am besten, Country am zweitbesten, und Age am schlechtesten abschneidet. Bei Gender bleibt nach Abzug der Personen, die nicht das gleiche Geschlecht angegeben haben, noch am meisten User übrig im Vergleich zu den anderen zwei DF-Methoden; also ist es auch eher so, dass mehr User mit sehr ähnlichem Musik-Geschmack in diesem User-Pool übrig geblieben sind. Bei Age ist dieser User-Pool am kleinsten, weil es im Vergleich wesentlich weniger User gibt, die das gleiche Alter haben; somit ist der User-Pool, von dem wir dann die ähnlichsten User herausnehmen, auch am kleinsten.

Zusammenfassend lässt sich sagen, dass wohl beide angeführten Erklärungsversuche Einfluss auf das erzielte Endergebnis haben.

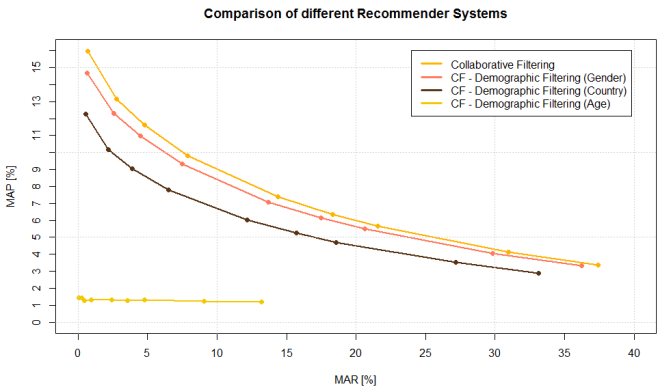

Figure 1: Vergleich von CF mit den drei DF-Arten

Anzahl der vorhergesagten Interpreten	Mean Average Precision (MAP)	Mean Average Recall (MAR)	F1 Score
1	14.66	0.63	1.21
5	12.30	2.57	4.26
10	10.98	4.51	6.40
20	9.32	7.47	8.29
50	7.09	13.70	9.34
75	6.13	17.50	9.08
100	5.49	20.65	8.67
200	4.06	29.85	7.15
300	3.34	36.23	6.12

Table 2: Performance des DF (Gender)

Anzahl der vorhergesagten Interpreten	Mean Average Precision (MAP)	Mean Average Recall (MAR)	F1 Score
1	1.44	0.06	0.11
5	1.42	0.28	0.46
10	1.29	0.48	0.70
20	1.32	0.96	1.11
50	1.31	2.44	1.71
75	1.27	3.56	1.88
100	1.30	4.81	2.05
200	1.23	9.09	2.17
300	1.20	13.18	2.20

Table 3: Performance des DF (Age)

Anzahl der vorhergesagten Interpreten	Mean Average Precision (MAP)	Mean Average Recall (MAR)	F1 Score
1	12.26	0.54	1.04
5	10.15	2.20	3.61
10	9.04	3.90	5.45
20	7.78	6.51	7.09
50	6.01	12.17	8.05
75	5.24	15.69	7.85
100	4.69	18.55	7.49
200	3.51	27.15	6.22
300	2.90	33.12	5.34

Table 4: Performance des DF (Country)

2.6 How are recall and precision related to each other for the different approaches and why?

Grundsätzlich lässt sich festhalten, dass bei einer anständigen Implementierung, die Präzision mit der Anzahl der Empfehlungen abnimmt, während der Recall steigt. Dieses Verhalten lässt sich anhand der Definitionen der beiden Größen gut erklären. Precision beschreibt das Verhältnis von Treffern zu empfohlenen Items. Je mehr Items wir empfehlen desto größer ist zwar die Chance auf einen Treffer, desto größer ist jedoch auch der Betrag im Nenner der Formel. Meist steigen die Treffer nicht in gleichem Maße an, weswegen die Precision mit zunehmenden Empfehlungen häufig abnimmt. Erklärt werden kann dieses Verhalten mit der zunehmenden Unsicherheit bei steigender Empfehlungszahl. Empfiehlt man nur wenige Items, so haben diese meist einen hohen Score und damit meist auch eine hohe Wahrscheinlichkeit unter den tatsächlich gehörten Items zu sein. Der Recall definiert sich dagegen aus dem Verhältnis von Treffern zu gehörten Items. Da mit zunehmendem Empfehlungen idR auch die Anzahl der Treffer steigen, die Anzahl der gehörten Items jedoch konstant bleiben, erhöht sich der Recall-Wert zunehmend. Um den Maximal möglichen Recall von 100% überhaupt erreichen zu können, müssen wir also mindestens soviele Items empfehlen, wie der Nutzer auch tatsächlich gehört hat.

Dieses Verhalten konnten wir bei all unseren implementierten Recommendern feststellen (vergleiche Abbildung 2). Eine Ausnahme bilden lediglich die Random-Baseline Varianten. Die gemessenen Precision-Werte bleiben auch bei variierender Empfehlungszahl relativ konstant, da sie ausschließlich durch den Zufall beeinflusst werden.

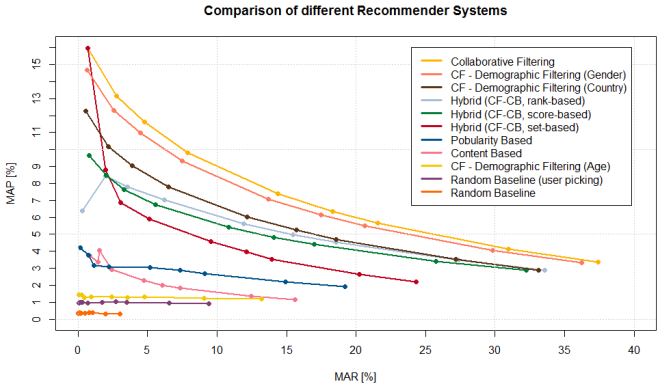

Figure 2: Vergleich aller benutzten Recommender Systeme anhand Mean Average Precision (MAP) und Mean Average Recall (MAR).

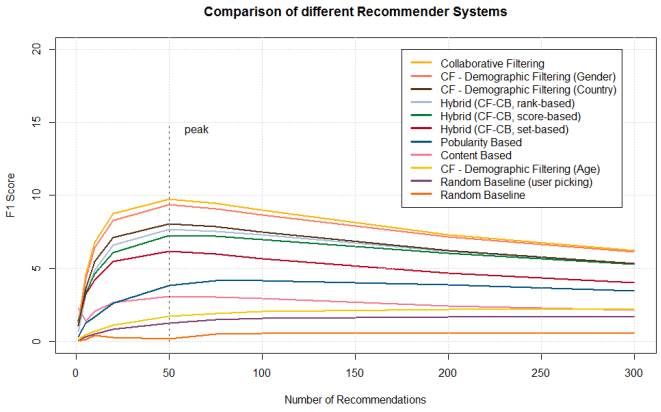

Figure 3: Vergleich aller benutzten Recommender Systeme anhand deren F1-Scores in Abhängigkeit der Anzahl der Vorhersagen.

In Abbildung 3 ist zu beobachten, dass der Peak bei 50 Number of Recommendations liegt. Selbstverständlich kann jedoch bei den Random Baselines kein Peak beobachtet werden, da die Precision bei diesen nicht mit Anzahl der Vorhersagen abfällt.

3. MÖGLICHKEITEN ZUR EVALUATION JENSEITS VON PRECISION UND RECALL

Precision und Recall alleine trifft keine Aussage über die Qualität der Reihenfolge der empfohlenen Items. Diese kann aber interessant sein, wenn man beispielsweise weiß, dass der Nutzer ohnehin nur die ersten X Items anschaut/anhört. Möglichkeiten die Reihenfolge zu evaluieren gibt es einige. Mit RR ließe sich angeben, an welcher Position der erste Treffer erfolgte. Mit P@K lässt sich angeben, wieviele Treffer es innerhalb der ersten K Empfehlungen gab. Daneben gibt es noch die Evaluation unter Klassifikationsaspekten wie z.B: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) und Rank Correlation.

Wenn die Situation es zulässt sind auch nutzerzentrierte Evaluationen in Erwägung zu ziehen: d.h. wir fragen die Nutzer über Fragebögen, Interviews etc., wie zufrieden sie mit den Vorschlägen sind. Beispielsweise lässt sich mit den berechneten Kriterien schlecht messen, wie neu, wie eintönig und langweilig, wie ähnlich die Lieder für den jeweiligen Nutzer waren; zum Beispiel ist es vielleicht eine gute Idee, Nutzern mehr Interpreten des Long Tails zu empfehlen, auch wenn uns eigentlich die Performance-Messung per MAP/MAR/F1-Score dafür abstrafen würden. Wir messen schließlich, ob der Nutzer die vorgeschlagenen Interpreten schon einmal gehört hat (d.h. im Test-Set vorkommen), und nicht, ob neue vorher unbekannte vorgeschlagene Interpreten dem Nutzer gefallen. Neben dem klassischen direkten Fragen könnte man die Nutzer auch einfach beobachten und deren Verhalten analysieren.