Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Eduard Weigandt

KDD mit implizitem Feedback

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Eduard Weigandt

KDD mit implizitem Feedback

Projektbericht eingereicht im Rahmen der Umsetzung von Projekt 1

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüfer: Prof. Dr. Kai von Luck

Eingereicht am: 15. Juni 2016

Eduard Weigandt

Thema der Arbeit
KDD mit implizitem Feedback

Stichworte
Analytik, Statistik, Matrix Factorization, Yoochoose

Kurzzusammenfassung

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Aufbau und Ziele 1

2 KDD mit Implizitem Feedback **2**
2.1 Charakteristika 2
2.2 Selektion von Daten 3
2.3 Vorverarbeitung 4
2.4 Transformation 6
2.4.1 Format 6
2.4.2 Features 7
2.5 Verfahren 8
2.5.1 Matrix Factorization 8
2.6 Evaluierung 10

3 Fazit und Ausblick **10**

1 Einleitung

In der vorherigen Arbeit Weigandt (2015) wurden erste Erfahrungen mit Algorithmen zur Erstellung von Empfehlungen gesammelt. Dafür wurde sich die *Matrix Factorization* in Verbindung mit unterschiedlichen Datensätzen[1] sowie die Implementierungen dieser in Lenskit und Mahout näher angeschaut. Für einen besseren Vergleich wurde zu diesem Zweck eine Toolchain[2] untersucht, die die Evaluierung von Empfehlungssystemen vereinfacht. Die dafür genutzten Datensätze lagen in Form von anonymisierten Nutzerbewertungen von Filmen und Büchern vor.

In der nun folgenden Projektarbeit geht es speziell um Nutzerverhalten und deren Charakteristika aus dem *Yoochoose* Datensatz (yoo, 2015), welcher für die *RecSys Challange 2015*[3] bereitgestellt wurde. Dieser Datensatz enthält jeweils Informationen über Aktionen von Nutzern einer E-Commerce Platform. Dabei sind die Aufzeichnungen in Form von Klickstrecken[4] der einzelnen Nutzer abgespeichert. Es existieren in diesem Fall keine Bewertungen wie in der ersten Projektarbeit, wo ausschließlich mit expliziten Bewertungen gearbeitet wurde. Im Forschungsfeld der Empfehlungssysteme ist explizites Feedback, die am meisten untersuchte und verwendete Art von Daten. Dabei darf man jedoch nicht die Psychologie hinter der Abgabe einer Bewertung vergessen. Menschen können durch mehrere Faktoren beeinflusst werden. So kann es passieren, dass zwei Menschen einen Filme für gut befinden, jedoch in einer Bewertung unterschiedliche Stimmen abgeben, weil von Person zu Person z.B. ein anderes Verständnis von gut und schlecht existiert. Mit implizitem Feedback versucht man diese möglichen Einflussfaktoren zu minimieren.

1.1 Aufbau und Ziele

Der Fokus in dieser Projektarbeit bezieht sich im Zusammenspiel von implizitem Feedback und den generierten Empfehlungen. Dafür werden unterschiedliche Methoden und Themen erläutert, die man dazu nutzen kann um Vorhersagen oder Empfehlungen zu erstellen. Die erläuterten Methoden und Fragen werden dann in einer weiterführenden Arbeit näher analysiert. Der Aufbau der hier vorliegenden Arbeit gestaltet sich wie folgt. Zuerst werden in Kapitel 2 die Besonderheiten von implizitem Feedback in Verbindung mit dem KDD (Fayyad u. a., 1996) Prozess vorgestellt. Dafür werden die besonderen Charakteristika dieser Datenart erläutert. Zum Schluss werden in Abschnitt 2.5 Methoden zur Verarbeitung des Datensatzes zum Zweck von Empfehlungen auf-

[1] *MovieLens* `http://grouplens.org/datasets/movielens/`, MovieTweetings `https://github.com/sidooms/MovieTweetings` und Book-Crossing `http://www2.informatik.uni-freiburg.de/~cziegler/BX/` (01.11.2015)

[2] RiVal `https://github.com/recommenders/rival` (01.11.2015)

[3] `http://recsys.yoochoose.net/challenge.html`

[4] Eine Klickstrecke besteht aus mehreren Aufzeichnungen von Klicks eines Nutzers, diese beinhalten jeweils einen Zeitstempel und den angeklickten Artikel.

gezeigt. Dafür wurden aktuelle Veröffentlichungen aus der Fachliteratur angeschaut. Die daraus resultierenden Ergebnisse sollen am Ende eine mögliche Roadmap skizzieren, wie man mit implizitem Feedback umgehen kann.

2 KDD mit Implizitem Feedback

2.1 Charakteristika

In Hu u. a. (2008) werden jeweils wichtige Aspekte vom impliziten Feedback erläutert, die bei der Erstellung von Empfehlungen für Fernsehzuschauern beobachtet wurden. Dabei wurde ein Verfahren aus dem Gebiet des *Collaborative Filtering* eingesetzt. Diese Klassen von Algorithmen benötigen als Eingabe die Historien[1] der Nutzer. Zudem ist diese Methode nicht an eine bestimmte Domäne gebunden. Das berechnete Model bezieht sich auf Beobachtungen von Nutzeraktionen in Verbindung mit einem Artikel. Die nachfolgenden Abschnitte fassen, die bisherigen Erkenntnisse übertragen auf den E-Commerce Bereich zusammen.

Kein negatives Feedback Wenn ein Nutzer einen Artikel kauft oder gut bewertet bedeutet, dass ein sehr hohes Interesse für diesen Artikel existiert. Im Umkehrschluss wiederum kann man nicht mit Bestimmtheit sagen, ob ein nicht gekaufter oder angeschauter Artikel dem Käufer nicht zusagt weil man zu wenig Informationen darüber besitzt. Der Kunde kennt vielleicht diesen Artikel noch gar nicht oder will erst zu einem späteren Zeitpunkt einen Kauf abwickeln. Nach Hu u. a. (2008) lässt sich in diesen fehlenden Informationen, das meiste negative Feedback entdecken.

Schmutzige- / Lückenhafte-Datensätze Ein weiterer Punkt, der die Verwendung erschwert, ist die Form der aufgezeichneten Daten eines Nutzers. Einen Film kann man schauen und danach schlecht bewerten, daraus kann man weitere Schlüsse für Empfehlungen ziehen. Bei einem gekauften Artikel ohne eine explizite Bewertung lassen sich jedoch schwer richtige Schlüsse ziehen. Das gleiche sieht man bei einer langen Verweildauer eines Kunden auf einer bestimmten Artikelseite. Der Nutzer muss nicht unbedingt ein Interesse am Artikel haben, sondern könnte z.Z. einfach nicht anwesend sein. Hierfür ist es wichtig eine Methode zu entwickeln, die solche Fälle ausschließen kann, um die Qualität der Daten zu erhöhen.

[1] Eine Aufzeichnung vom Nutzerverhalten in Form von Kaufhistorien, Gewohnheiten oder Aktivitäten.

Grad des Vertrauens in die Beobachtung Aufbauend auf den zwei vorherigen Aspekten sieht man, dass eine definitive Aussage für eine Vorliebe zu einem Artikel nicht getroffen werden kann, sondern nur eine Einschätzung für die Interpretation der Beobachtung. Das bedeutet z.B. je häufiger ein Artikel angeschaut wurde, desto höher ist die Wahrscheinlichkeit für das Interesse an diesem. Dies trifft wiederum nicht immer für alle Artikel zu, da man sich vielleicht alltägliche Gebrauchsgüter öfter bestellt, diese jedoch sind mit dem eigenen Lieblingsbuch oder Film nicht gleich zu setzen. In Unterabschnitt 2.4.2 werden weitere Merkmale, die man nutzen kann um eine Relevanz für einen Artikel zu berechnen aufgeführt.

Metriken & Evaluierung Es gibt keine klaren Metriken, die eine Empfehlung anhand des impliziten Feedbacks bewerten können. Die Rahmenbedingungen sind, wie man anhand der ersten drei Punkte erkennen kann nicht eindeutig. Jedoch kann man anhand von Dauer eines Aufenthalts und dem letztendlichen Kauf eines Artikels auf einen neuen möglichen Kauf schließen. Je nachdem was die Priorität der E-Commerce Plattform sind, kann ein empfohlener und danach auch gekaufter Artikel am entscheidendsten sein. Eine klarere Aussage könnte man mit explizitem Feedback bekommen, falls der Nutzer zusätzlich eine Bewertung abgeben würde.

2.2 Selektion von Daten

Die im vorherigen Abschnitt 2.1 genannten Charakteristika von implizitem Feedback müssen beim KDD Prozess berücksichtigt werden. Dafür werden in den folgenden Abschnitten mögliche Methoden sowie Strategien vorgestellt und diskutiert.

Domäne Wie in der Einleitung erwähnt wird ein fertiger Datensatz von yoo (2015) verwendet, welcher für die *RecSys Challange 2015* bereitgestellt wurde. Die Ziele im Wettbewerb bestanden aus den folgenden zwei Punkten:

1. Is the user going to buy items in this session? (Yes / No)
2. If yes, what are the items that are going to be bought?

Zur Beantwortung der oberen Fragen wurden jeweils zwei unterschiedliche Informationsquellen bereitgestellt. Einmal ein Datensatz der das Nutzerverhalten von Kunden auf der Seite in Form von Klicks auf Artikel und deren Bestellungen beinhaltet. Sowie ein weiterer Datensatz der zur Bestreitung des Wettbewerbs zusätzliche Nutzeraufzeichnungen, jedoch ohne Bestellungen, beinhaltet. Dieses Testset ist für die Bewertung der eingesetzten Methoden am Ende des Wettbewerbs gedacht.

Grund für die Wahl Zu einem bietet dieser Datensatz die Möglichkeit einen guten Vergleich zwischen vielen unterschiedlichen Ansätzen aus dem Wettbewerb zu ziehen. Zum anderen gibt es nur wenige wissenschaftliche Arbeiten, die diese Art des Feedbacks untersuchen. Da solche Informationen meistens aus Datenschutzstechnischen Gründen nicht herausgegeben werden. Aus diesem Grund ist es von Interesse, zu den vorherigen auch die folgenden Fragen zu beantworten:

1. Reicht das implizite Feedback aus, um gute Vorhersagen über das Nutzerverhalten zu erstellen?
2. Welche Verfahren eignen sich am besten für diese Art der Daten?

2.3 Vorverarbeitung

Durch den Einsatz eines Datensatzes aus einem Wettbewerb besteht im Vorfeld eine gute Qualität der Daten, unter der Annahme das keine Fehler bei der Aufzeichnung oder Aufbereitung der Daten passiert sind. In der Realität is dies jedoch nicht immer der Fall. Die größten Probleme, die dann auftreten, sind einmal das Fehlen von z.B. Bewertungen oder Datenredundanz, welche die Berechnung eines Modells unnötig verlängert (Sar Shalom u. a., 2015). Zudem können auch fehlerhafte Datensätze durch falsche Bedienung oder Fehler im System hinzukommen. Dies sind nur ein paar Qualitätsmerkmale, die man je nach Datensatz priorisieren muss. Weitere gute Beispiele lassen sich in Wang u. Strong (1996) finden.

Fehlende Daten

Beim Kalt-Start Problem existieren anfangs nur wenige bis gar keine Bewertungen von Artikeln. Häufig werden diese Artikel für das Modell nicht mit einberechnet. Ein weiterer Aspekt, den man nicht unterschätzen darf, liegt in den sehr lückenhaften Informationen in den Daten. Wenn man z.B. ein sehr großes Angebot an Artikeln besitzt, kann es sein das sich die Artikel von Kunden nicht überschneiden und somit schwieriger Ähnlichkeiten gefunden werden können. In Abschnitt 2.1 wird darauf hingewiesen, dass das nicht Ansehen von Artikeln keine eindeutige Aussage für mangelndes Interesse an diesen ist. In Ostuni u. a. (2013) wird eine Methode vorgestellt, die die einzelnen Pfade und Beziehungen in einem bipartiten Graphen (siehe Abbildung 2.1) dazu

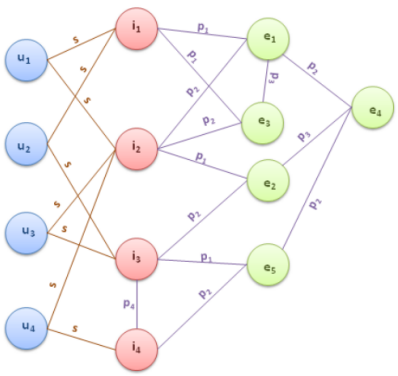

Abbildung 2.1: (Quelle: Ostuni u. a. (2013))

nutzt, um ein Top-N Ranking von Empfehlungen mit impliziten Informationen zu berechnen. Übertragen auf den Yoochoose Datensatz stellen die einzelnen Kategorien, Käufe und Klicks in den Nutzerhistorie die Relationen dar, um die relevanten Pfad-basierten Features in der Klickstrecke zu berechnen. Dadurch werden die unbekannten Daten anhand der genannten Beziehung zu den bekannten eingeordnet.

Redundanz in den Daten

Auf einer Online-Vertriebsplattform wie z.B. Amazon gibt es je nach Land mehr als 100 Millionen Artikel, welche redaktionell oder automatisch von Händlern bereitgestellt werden. Dabei kann ein und derselbe Artikel mehrfach von unterschiedlichen Händlern angeboten werden. Für eine effiziente Berechnung eines Modells für Empfehlungen muss man solche Einträge zusammenfassen. In Abbildung 2.2 sieht man ein Suchergebnis für ein Kartenspiel, welches bei dem einen Händler übersetzt wurde und bei dem anderen nicht[2].

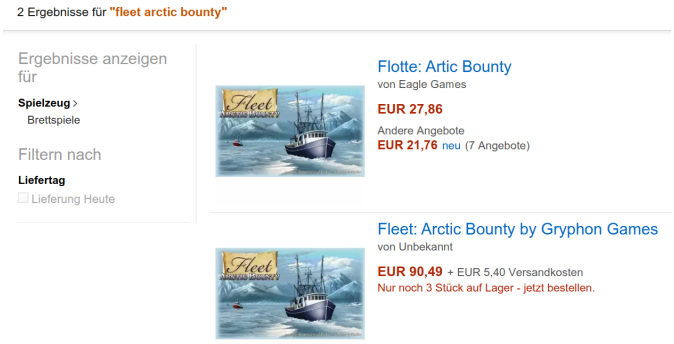

Abbildung 2.2: Suchergebnis für das gleiche Kartenspiel mit redaktionellen Fehlern. (Quelle: amazon.de)

Da der hier verwendete Datensatz in einer anonymisierten Form vorliegt ist es schwierig bis unmöglich auf solche Fehler einzugehen, da man diese nicht alleine über die Artikelnummern erkennen kann.

Nutzermodell

In den meisten Fällen möchte man Ähnlichkeiten in den Nutzerhistorien finden, um daraus Empfehlungen für diese Nutzer abzuleiten. Wenn man jedoch eine sehr hohe Dichte an Daten besitzt, kann sich die Berechnung in die Länge ziehen. Das von Sar Shalom u. a. (2015) vorgeschlagene Verfahren für diesen Fall besteht in der richtigen Zusammenfassung (Sampling) dieser Datensätze, so dass am Ende ein gleich gutes Modell herauskommt, ohne zu viele Informationen zu

[2] Es existiert nur die englische Version von diesem Spiel.

verlieren. In den vorgestellten Experimenten wurden damit gute Ergebnisse erzielt, jedoch darf man nicht vergessen, dass die Bewertungen von Nutzern sich über die Zeit verändern (Amatriain u. a., 2009). Dadurch ist es fraglich, wie weit man mit solchen Optimierungs-Schritten arbeiten sollte. Als eine Verbesserung hierfür wird in Abschnitt 2.5 eine Alternative vorgestellt, welche eine inkrementelle Aktualisierung des Datenmodells ermöglicht, um den Wandel von Vorlieben mit ein zu beziehen.

2.4 Transformation

In diesem Abschnitt wird das Thema der Datenverdichtung behandelt, dazu gehört das Definieren von Features zur Reduktion des Problemraums. Zuerst wird jedoch das vorhandene Format des Datensatzes näher beleuchtet.

2.4.1 Format

Im Trainingsset sind jeweils zwei Dateien vorhanden, die unterschiedliche Arten von Informationen enthalten (Ben-Shimon u. a., 2015). In der ersten Datei sind alle Klickstrecken der Nutzer gespeichert. Das Format dieser Datei sieht man in Tabelle 2.1. Insgesamt enthält der gesamte Datensatz 33,040,175 Einträge aus der ersten Datei und 1,177,769 Einträge aus der zweiten (siehe Tabelle 2.2). Insgesamt sind zusammengefasst 9,512,786 Sessions in der 1gb großen Datei enthalten, die über den Zeitraum von sechs Monaten aufgezeichnet wurden. Von diesen Sessions wurden in ca. 5 % der Fälle ein Kauf getätigt.

Session ID	Zeitstempel ID	Artikel ID	Kategorie
2	2014-04-07T14:02:36.889Z	214551617	0
3	2014-04-02T13:17:46.940Z	214716935	0
3	2014-04-02T13:26:02.515Z	214774687	0
4	2014-04-07T12:09:10.948Z	214836765	0

Tabelle 2.1: Nutzerhistorie über Klickstrecke

Außerdem geht aus den Daten hervor, dass am Anfang der Woche die wenigsten Käufe und am Wochenende die meisten getätigt wurden. Dies wird auch von den meisten Quellen (z.B. Lieferdienste) so bestätigt. In Romov u. Sokolov (2015) wurden die Sessions mit den meisten Klicks zu dem als die wahrscheinlichsten für einen Kauf identifiziert.

Session ID	Zeitstempel	Artikel ID	Preis	Anzahl
420374	,2014-04-06T18:44:58.314Z	214537888	12462	1
420374	2014-04-06T18:44:58.325Z	214537850	10471	1
281626	2014-04-06T09:40:13.032Z	214535653	1883	1

Tabelle 2.2: Kaufhistorie der Nutzer

Fazit Das Format des Datensatzes ist klar und enthält alle wichtigen Informationen zu einem Bestellungsprozess, jedoch birgt die schiere Menge an Daten ein potentielles Risiko bei der Berechnung von Empfehlungen. Das CSV-Format ist redundant und musste deswegen in eine SQL-Datenbank überführt werden, damit die teilweise Berechnung von statistischen Werten im Arbeitsspeicher möglich wurde. Für weiterführende Berechnungen muss eine performante Lösung überlegt werden, die das Volumen bewältigen kann.

2.4.2 Features

Je nach Art einer Methode zur Erstellung von Empfehlungen werden Gemeinsamkeiten bzw. Muster im Verhalten zu bestimmten Artikeln in einer Session gesucht. Diese Merkmale werden auch Features genannt und enthalten Informationen über Eigenschaften des Datensatzes. Features werden je nach Algorithmus entweder berechnet oder vorher festgelegt. Bei der Matrix Factorization kann man z.B. den zerlegten Wert aus den unterschiedlichen Features in den jeweiligen Matrizen zusammen setzen lassen. Dies wurde in der vorherigen Arbeit näher beschrieben. (Weigandt, 2015)

RecSys Challange Gewinner Der Gewinner Romov u. Sokolov (2015) aus der *RecSys Challange 2015* liefert einen Ansatzpunkt (siehe Tabelle 2.3) für relevante Features aus dem *Yoochoose* Datensatz. Dabei haben die Autoren jeweils zwei Kategorien von Features benannt, die aus ihrer Sicht die besten Ergebnisse liefern. Zu diesen zählen einmal die Session sowie die Session-Item-basierten Features. Die ersteren beziehen sich auf die Kerndaten einer Session (Dauer des Aufenthalts, Anzahl an Klicks, etc.) und die letzteren auf die Beziehung zwischen der Session und einem oder mehreren Artikel.

Fazit Auch wenn schon erfolgreiche Lösungen für diesen Datensatz existieren, ist es von Interesse neue und bekannte Verfahren gegenüber zu stellen und zu vergleichen. Darüber hinaus bleibt eine weitere Frage (siehe Abschnitt 2.2) offen: Welche der am Ende verwendeten Features sind wirklich für das Endergebnis ausschlaggebend?

Session:

- Dauer der Session
- Anzahl an Klicks, einzigartige Artikel / Kategorien und Kategorie-Artikel Paare
- Top 10 Artikel / Kategorien
- ...

Session-Item:

- Artikel ID in Kategorie
- Anzahl von Ersten- / Letzten-Klick auf einen Artikel in einer Kategorie (Monat, Tag, Monat-Tag, etc.)
- Zeit vor dem Klick und nach dem Klick auf einen Artikel
- Anzahl an einzigartigen Kategorien in der Session des Artikels
- ...

Tabelle 2.3: Übersetzer Auszug von Features aus Romov u. Sokolov (2015)

2.5 Verfahren

In diesem Abschnitt werden weitere Aspekte vom Verfahren vorgestellt, die in Kombination genutzt werden, um aus dem vorliegenden Datensatz neue Informationen zu extrahieren.

2.5.1 Matrix Factorization

Eine der ersten Arbeiten, die sich mit implizitem Feedback auseinandersetzten, findet man bei Hu u. a. (2008). Die meisten größeren Implementierungen (siehe Apache Spark[3]) basieren auf diesem Verfahren. In den nachfolgenden Abschnitten werden mögliche Erweiterungen für dieses vorgestellt.

Top-N Ranking Evaluierung Eine mögliche Variante um Empfehlungen zu berechnen besteht in der Berechnung von Bewertungen von Artikeln anhand von bestehenden Bewertungen anderer Nutzer. Eine weitere findet man in der Suche einiger bestimmter Artikel aus der Menge, die eine hohe Relevanz für den Nutzer haben. Bei dieser Variante werden die Top-N Empfehlungen mit Hilfe von Ranking Algorithmen gesucht. Diese machen eine Punktevergabe auf dem Datenmodell, die man zur Sortierung nutzen kann.[4]

In Steck (2010) wird dafür ein Datenmodell in Verbindung mit einem Verfahren zur Performance Messung Lim u. a. (2015) (**Average Discounted Gain**) vorgestellt. Die Relevanz zu einem Artikel wird dabei in binärer Form dargestellt und durch eine Optimierung in Form eines

[3] `https://spark.apache.org/docs/latest/mllib-collaborative-filtering.html`
[4] Ein Beispiel für solch eine Top-N Ranking wird in Abschnitt 2.3 Fehlende Daten referenziert.

Minimierungs-Problem gelöst. Die verwendeten Daten bei den Experimenten zur Evaluierung bestanden dabei aus aufbereiteten expliziten Bewertungen aus der Kategorie *Video Games* von Amazon, den Musikgewohnheiten aus last.FM[5] und den Filmbewertungen von MovieLens[6]. Dabei wurden Bewertungen von vier bis fünf Sternen und Lieder, die mindestens drei Mal gespielt wurden, als relevant für den Nutzer eingestuft. Die Art der Daten ist in diesem Fall nicht sehr optimal, da nicht ausschließlich implizite Daten genommen wurden, sondern eine ausgesuchte Interpretation dieser.

Fazit Das vorgeschlagene Verfahren eignet sich, um die Performanz von weiteren Top-N Empfehlungssystemen auf großen Datensätzen zu überprüft, da ein effizientes Sampling der Daten stattfindet.

Dynamisches Modell Wie schon in den vorherigen Abschnitten erwähnt, wandeln sich Vorlieben von Nutzern über die Zeit hinweg. Um diese Veränderung zu erfassen kann man nach einer bestimmten Anzahl an neuen Informationen, dass komplette Datenmodell oder nur einzelne Matrizen neu berechnen. In Abbildung 2.3 sieht man im Gegnsatz dazu den Ablauf einer Aktualisierung, die sich inkrementell über die Zeit erstreckt. Dafür wird in Song u. a. (2015) eine Erweiterung für die Matrix Factorization vorgeschlagen, um die gleichbleibende Genauigkeit des Modells sicherzustellen.

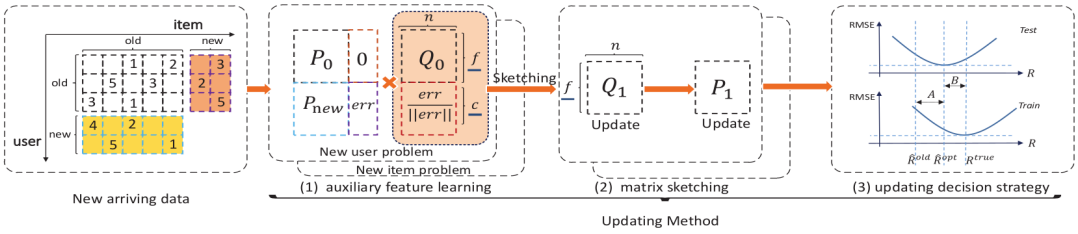

Abbildung 2.3: (Quelle: Song u. a. (2015))

Ablauf einer inkrementellen Aktualisierung des Datenmodells:

1. Berechnung der einzelnen Fehler-Abweichungen zu den bekannten Daten. Danach werden diese normalisiert und zu den bestehenden Featurs als weiterer Faktor hinzugefügt.
2. Verkleinerung der erhöhten Featuranzahl durch *Low Rank Matrix Sketching*, welches eine Art von Sampling für Datenströme darstellt.
3. Anpassung der Strategie für weitere Aktualisierungen des Modells. Unnötige inkrementelle Veränderungen, die keine Verbesserung in den Vorhersagen führen, werden abgelehnt.

[5] http://www.last.fm/de/
[6] http://grouplens.org/datasets/movielens/

Die Vorteile eines solchen Prozesses liegen in der langfristigen Genauigkeit und dem Vermeiden einer kompletten Neuberechnung. Ein Nachteil besteht nach Song u. a. (2015) in einer möglichen Überanpassung an die Daten. Zudem muss man sagen, dass die eingesetzten Datensätze ausschließlich Bewertungen von Artikeln enthielten und keine Nutzerinteraktionen mit den Seiten.

2.6 Evaluierung

Die Bewertung der einzelnen Verfahren wird jeweils durch das mitgelieferte Trainingsset in Verbindung mit einem Lösungsset, wo die Käufe aus dem Trainingsset enthalten sind, ermöglicht. Der größte Teil des Datensatzes wird somit zum Anlernen des Modells genutzt und das Trainingsset wiederum für die Experimente. Die Güte der Empfehlungen wird dann anhand der am Ende gekauften Produkte aus dem Lösungsset berechnet. Für die *RecSys Challange 2015* wurde zu diesem Zweck die nachfolgende Formel bereitgestellt.

$$Score(\boldsymbol{Sl}) = \sum_{\forall s \in \boldsymbol{Sl}} \begin{cases} if\ s \in \boldsymbol{Sb} & \rightarrow \frac{|S_b|}{|S|} + \frac{|A_s \cap B_s|}{|A_s \cup B_s|} \\ else & \rightarrow -\frac{|S_b|}{|S|} \end{cases}$$

Abbildung 2.4

Die einzelnen Variablen schlüsseln sich wie folgt auf:

- Sl – Sessions in der eingereichten Lösungsdatei.
- S – Alle Sessions im Testset.
- s – Eine Session auf dem Testset.
- Sb – Session aus dem Testset, welche mit einem Kauf endete.
- As – Vorgeschlagenen und gekaufte Artikel in der Session s.
- Bs – Tatsächlich gekauften Artikel in der Session s.

3 Fazit und Ausblick

Wie schon in der Einleitung erwähnt, werden die meisten Untersuchungen mit expliziten Angaben wie z.B. einer Bewertung durchgeführt. Zum einen bieten solche Datensätze eine direktere Form von Vorlieben eines Benutzers. Zum andere muss, dass jedoch nicht immer der Wahrheit entsprechen, weil sich die Vorlieben eines Menschen über die Zeit verändern. Hinzu kommen

noch absichtliche oder unabsichtliche Falschangaben. Aus diesem Grund muss das Datenmodell flexibel genug sein, um diese zeitlichen Veränderungen zu berücksichtigen.

Beim implizitem Feedback wiederum gibt es keine direkte negative Bewertung von Artikeln da man nur Beobachtungen des Nutzerverhaltens besitzt. Dadurch muss man die Beobachtung mit einem Grad an Vertrauen ausstatten. In der Domäne des E-Commerce sind Käufe die ausschlaggebendste Information, ob ein empfohlener Artikel gut oder schlecht ist. Auch hier können Fehler in den Daten auftreten in Form von fehlenden oder redundanten Daten, jedoch ist so was bei anonymisierten Daten nicht immer ersichtlich.

Die Qualität bzw. Anzahl der zu Forschungszwecken verfügbaren Datensätze unterscheidet sich stark. Aus diesem Grund werden in einigen Arbeiten die Experimente mit ungenügenden oder aufbereiteten Daten umgesetzt. Zu dieser Situation tragen viele Unternehmen bei, da solche Art von Informationen als Firmengeheimnisse eingestuft werden. Denn darin lassen sich viele persönliche Daten der Nutzer sowie neue Möglichkeiten um Geld durch bessere Empfehlungen zu verdienen, wiederfinden.

Der in dieser Arbeit vorgestellte KDD Prozess dient dazu um weiterführende Arbeiten darauf aufzubauen. Es sind im Verlauf mehrere wichtige Fragen aufgestellt worden, die sich mit den speziellen Merkmalen der gezeigten Daten auseinander setzen. Dadurch kann man einen neuen Blickwinkel zum Thema des implizitem Feedback einnehmen, welcher durch den Gewinner der *RecSys Challange 2015* noch nicht untersucht wurde. Dazu zählt z.B. eine dynamische Berechnung des Datenmodells welches das Wegfallen einer unnötige Neuberechnung aller Daten, sowie die schnellere Reaktion auf das aktuelle Kundenverhalten, ermöglicht.

Literaturverzeichnis

[yoo 2015] *YOOCHOOSE e-commerce data set - (Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License)*. `http://recsys.yoochoose.net/challenge.html`, 2015 1, 3

[Amatriain u. a. 2009] Amatriain, Xavier ; Pujol, Josep M. ; Tintarev, Nava ; Oliver, Nuria: Rate It Again: Increasing Recommendation Accuracy by User Re-rating. In: *Proceedings of the Third ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2009 (RecSys '09). – ISBN 978–1–60558–435–5, 173–180 6

[Ben-Shimon u. a. 2015] Ben-Shimon, David ; Tsikinovsky, Alexander ; Friedmann, Michael ; Shapira, Bracha ; Rokach, Lior ; Hoerle, Johannes: RecSys Challenge 2015 and the YOOCHOOSE Dataset. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15). – ISBN 978–1–4503–3692–5, 357–358 6

[Fayyad u. a. 1996] Fayyad, Usama ; Piatetsky-Shapiro, Gregory ; Smyth, Padhraic: From data mining to knowledge discovery in databases. In: *AI magazine* 17 (1996), Nr. 3, S. 37 1

[Hu u. a. 2008] Hu, Yifan ; Koren, Y. ; Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In: *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, 2008. – ISSN 1550–4786, S. 263–272 2, 8

[Lim u. a. 2015] Lim, Daryl ; McAuley, Julian ; Lanckriet, Gert: Top-N Recommendation with Missing Implicit Feedback. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15). – ISBN 978–1–4503–3692–5, 309–312 8

[Ostuni u. a. 2013] Ostuni, Vito C. ; Di Noia, Tommaso ; Di Sciascio, Eugenio ; Mirizzi, Roberto: Top-N Recommendations from Implicit Feedback Leveraging Linked Open Data. In: *Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2013 (RecSys '13). – ISBN 978–1–4503–2409–0, 85–92 4

[Romov u. Sokolov 2015] Romov, Peter ; Sokolov, Evgeny: RecSys Challenge 2015: Ensemble Learning with Categorical Features. In: *Proceedings of the 2015 International ACM Recommender Systems Challenge*. New York, NY, USA : ACM, 2015 (RecSys '15 Challenge). – ISBN 978–1–4503–3665–9, 1:1–1:4 6, 7, 8

[Sar Shalom u. a. 2015] Sar Shalom, Oren ; Berkovsky, Shlomo ; Ronen, Royi ; Ziklik, Elad ; Amihood, Amir: Data Quality Matters in Recommender Systems. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15). – ISBN 978–1–4503–3692–5, 257–260 4, 5

[Song u. a. 2015] Song, Qiang ; Cheng, Jian ; Lu, Hanqing: Incremental Matrix Factorization via Feature Space Re-learning for Recommender System. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA : ACM, 2015 (RecSys '15). – ISBN 978–1–4503–3692–5, 277–280 9, 10

[Steck 2010] Steck, Harald: Training and Testing of Recommender Systems on Data Missing Not at Random. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA : ACM, 2010 (KDD '10). – ISBN 978–1–4503–0055–1, 713–722 8

[Wang u. Strong 1996] Wang, Richard Y. ; Strong, Diane M.: Beyond Accuracy: What Data Quality Means to Data Consumers. In: *J. Manage. Inf. Syst.* 12 (1996), März, Nr. 4, 5–33. http://dx.doi.org/10.1080/07421222.1996.11518099. – DOI 10.1080/07421222.1996.11518099. – ISSN 0742–1222 4

[Weigandt 2015] Weigandt, Eduard: Toolchain zur Untersuchung von Empfehlungssystemen basierend auf Matrix Factorization. In: *Projekt 1* (2015) 1, 7