

Task 3: Paper Zusammenfassung

*Exploiting User Interests for Collaborative Filtering:
Interests Expansion via Personalized Ranking*

Autoren: Qi Liu, Enhong Chen, Hui Xiong, Chris H.Q. Ding

Paper verfügbar unter: <http://dm.ustc.edu.cn/docs/2010/p1697-liu.pdf>

Anmerkungen zum Paper

Diese Arbeit wurde gewählt, da sie einen interessanten Ansatz zur Erstellung von Vorschlägen von Produkten für Benutzerinnen und Benutzer behandelt. Jener Algorithmus geht bei der Berechnung der Vorschläge andere Wege als die meisten sonstigen Recommender-Systeme (RCS). Beschreibung, Charakteristika und Anmerkungen sollen in der nachfolgenden Zusammenfassung behandelt werden.

Zusammenfassung

1. Allgemeines

Kollaborative Filtertechniken sind bereits sehr gut erforscht und finden viele verschiedenste Einsatzgebiete im Bereich der RCS, da diese Art Systeme nur Informationen über User-Interaktionen benötigen. Diese Systeme erforschen und lernen allerdings nur Beziehungen zwischen User und Items und vernachlässigen die Tatsache, dass das Userverhalten stark von (verborgenen) Interessen beeinflusst wird und produzieren somit teils ungenaue Resultate. Des Weiteren ändern sich diese Userinteressen von Zeit zu Zeit, was traditionelle kollaborative Filtersysteme allerdings nicht abbilden können. Dies führt zu „Überspezialisierung“ der Vorschläge und somit zu weniger Nutzen für den User. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.1]

Der in diesem Paper behandelte Ansatz führt eine weitere Schicht zwischen die reine User-Item-Beziehung ein, nämlich die Interessen der User. Im Speziellen wird ein Interesse als Anforderung betrachtet, um vom User auf Items zu kommen, während beim dazugehörigen Item das Interesse als eine seiner Eigenschaften betrachtet wird. Es ist laut dieser Abhandlung wichtig, sich dieses drei-Schichten-Modell zu Nutze zu machen, da es zu besser interpretierbaren Vorschlagsresultaten führen und die Interaktionen von Usern, Items und User-Interessen verständlicher machen soll. [(Liu, Chen, Xiong, & Ding, 2010) S.1]

Um die oben geschilderten Probleme anzugehen, wird in diesem Paper der Algorithmus *iExpand* vorgestellt, der in Nachfolgendem Kapitel beschrieben werden soll.

2. iExpand

Die allgemeine Funktionsweise von *iExpand* stellt sich wie folgt dar:

- Jedes User-Interesse wird zuerst durch einen Faktor erfasst.
- Diese Interessen werden extrahiert und die Übergangsmöglichkeiten zwischen verschiedenen Interessen werden erlernt.
- Die möglichen Interessenserweiterungen (neue Interessen) von Usern werden durch personalisiertes Ranking modelliert, d.h. die nächsten möglichen Interessen eines Users können vorhergesagt werden.

[vgl. (Liu, Chen, Xiong, & Ding, 2010) S.1]

Durch die Interessenserweiterung (interests expansion) können vielfältigere Empfehlungen gegeben werden, was der Überspezialisierung entgegensteuert. Die einzelnen Schritte des Algorithmus werden in den folgenden Unterkapiteln etwas genauer erläutert.

2.1. User-Interessen

Die Information über die verborgenen Interessen von Usern beinhaltet die Wahrscheinlichkeitsverteilung von jedem User über jedes Interesse, die Wahrscheinlichkeitsverteilung von jedem Interesse über jedes Item, sowie die Verteilung jedes Interesses. Diese Matrizen und Vektoren werden durch die *Gibbs sampling* Technik aus dem *LDA*-Modell extrahiert. *LDA* ist ein themenbasiertes Ähnlichkeitsmodell, von dem iExpand die Interessen extrahiert, was aber anders als in topic models nicht themenzentrisch geschieht. Bei iExpand hat jeder User eine Verteilung auf das gesamte Interessensspektrum, während beim PLSA/LDA die Verteilungen allein durch das Topic (Thema) spezifiziert werden. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.2]

Wahrscheinlichkeitsverteilung von User über Interesse:

Verteilung von User U_i über Interesse T_j ist θ_{ij}

Wahrscheinlichkeitsverteilung von Interesse über Item:

Verteilung von Interesse T_j über Item I_i ist ϕ_{ij}

Verteilung eines Interesses: $\vec{\vartheta}_i$

2.2. User-Interessen Korrelationsgraph

Dieser Graph repräsentiert die Übergangswahrscheinlichkeiten zwischen den Interessen. Um diesen zu konstruieren, werden die Items als Zwischenentitäten verwendet. Das bedeutet, dass die Interessenszusammenhänge anhand der Items und Item-Charakteristiken in einer Matrix (φ) zusammengefasst werden. Formel:

$$\varphi_{ij} = P(T_j|I_i) = \frac{P(T_j, I_i)}{P(I_i)} = \frac{\phi_{ij}\vec{\vartheta}_j}{\sum_{k=1}^K \vec{\vartheta}_k \phi_{ik}}$$

Die Korrelationen zwischen Interessen werden in Form von Wahrscheinlichkeiten modelliert. Dazu wird zuerst der bipartite Graph $G = \langle X, E \rangle$ gebildet, um die Beziehungen zwischen Items und Interessen abzubilden (mit Knotenmenge $X = I \cup T$). Das Gewicht der Kante von Interesse T_j nach Item I_i ist ϕ_{ij} und das

Gewicht der Kante von I_i nach T_j ist φ_{ij} . Nach der Projektion von G ist die Beziehungsmatrix zwischen Interessen (ψ) vorhanden, wobei ψ_{ij} die Empfehlungsstärke des Interesses T_i für T_j angibt. Dies kann durch die folgende Formel berechnet werden:

$$\psi_{ij} = P(T_j|T_i) = \sum_{n=1}^N P(T_j|I_n)P(I_n|T_i) = \sum_{n=1}^N \varphi_{nj} \Phi_{ni}$$

Der bipartite Graph wird schließlich in einen Korrelationsgraphen umgewandelt, der die Beziehungen zwischen Interessen beschreibt, wobei ψ die Korrelationsmatrix darstellt. In der Korrelationsmatrix beschreibt ψ_{ij} den Korrelationskoeffizienten zwischen T_i und T_j aus der Sicht von T_i . [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.2]

2.3. User-Interessenserweiterung

In diesem Schritt werden User-Interessen personalisiert gereiht, was auf den User-Interessen-Korrelationsgraphen angewandt wird. Dazu wird der *PageRank*-Algorithmus wiederholt auf den Interessenvektor eines Users angewandt bis Konvergenz erreicht wird. Der finale Vektor enthält dann die erweiterten User-Interessen, was auch als Vorhersage des nächsten möglichen Interesses eines Users betrachtet werden kann. Aufgrund dieses Ansatzes sollen vielfältige Empfehlungen systematisch möglich sein. Algorithmisch wird hier an der personalisierten Reihung angesetzt, welche mit einem *random walk with restart (RWR)* über dem Graphen für jeden User durchgeführt wird, bis Konvergenz erreicht wird. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.2]

2.4. Optimale Empfehlungen

iExpand reiht Items nach der Relevanz für jeden betrachteten User, wobei die Verteilung des Users auf Interessen als Zwischenentität fungiert. Empfehlungslisten können einfach durch die Reihung der infrage kommenden Items generiert werden, wobei der Parameter K die Anzahl der Empfehlungen angibt. Das bedeutet wiederum, dass Empfehlungen ohne die Vorhersage von Bewertungen getroffen werden können.

Formel:

$$P(I_j|U_i) = \sum_{k=1}^K P(I_j|t = k)P_s(t = k|U_i) = \sum_{k=1}^K \Phi_{jk}\theta_{ik}^{(s)}$$

iExpand kann des Weiteren auch als Algorithmus zur Vorhersage von Bewertungen verwendet werden. Auf die Formel wird hier nicht näher eingegangen, da dieses Verfahren eigentlich ohne Bewertungsvorhersage auskommt.

Um die Performance des Prozesses zu wahren, können der Folgerungsprozess und der Korrelationsgraph periodisch offline aktualisiert werden, während die User-Interessen aktualisiert werden, wenn eine Bewertung des Users abgegeben wird. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.2, 3]

2.5. Abschätzen der Parameter

Für das Funktionieren des Algorithmus *iExpand* ist das Abschätzen der Bayes'schen Hyperparameter α und β , sowie der Interessenanzahl K vonnöten. Dies wird durch einen Algorithmus erreicht, der diese Parameter „lernt“. Für alle in Frage kommenden Werte für K werden Startwerte für α und β festgelegt und iterativ nach einem *Gibbs sampling* laut den im Paper angegebenen Formeln aktualisiert. Nach diesem Aktualisierungsvorgang werden die Parameter durch die sog. *Chib-style estimation* abgeschätzt und im Anschluss an alle Durchgänge die besten Werte gewählt. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.3]

3. Experiment und Ergebnisse

Um die Effektivität des *iExpand*-Algorithmus darzulegen, wurde dieser im Falle seiner Verwendung als Reihungsalgorithmus mit den bereits existierenden Systemen ItemRank, L^+ , sowie LDA und SVD verglichen. Als Vergleichswerkzeuge wurden hier *Degree of Agreement* (DOA) und *Hit Ratio* (HR) verwendet. Des Weiteren wurden verschiedene Aufteilungen der Daten in Trainingsdaten und tatsächliche Messdaten verglichen. *iExpand* schnitt bei diesen Messungen immer besser ab als die Konkurrenz, vor allem bei den Tranchen mit wenig Trainingsdaten. Je mehr

Trainingsdaten verwendet wurden, desto geringer wurde der Unterschied der Ergebnisse verglichen mit LDA. [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.3, 4]

In der Verwendung als Bewertungsalgorithmus wurde er mit UCF, RSVD, LDA und ItemRank verglichen. Um diese Ergebnisse zu vergleichen zu können, wurden die beiden Fehlermessungen *Mean Absolute Error* (MAE – Mittlerer absoluter Fehler) und *Root Mean Squared Error* (RMSE – Wurzel des mittleren quadratischen Fehlers) verwendet. Hier waren die Messergebnisse von *iExpand* nur bei den beiden Tranchen mit den geringsten Mengen an Trainingsdaten besser als die der Konkurrenz. Dies lässt sich wohl darauf zurückführen, dass *iExpand* in seiner eigentlichen Verwendung kein Algorithmus zur Vorhersage von Bewertungen ist [vgl. (Liu, Chen, Xiong, & Ding, 2010) S.3, 4]