

2021 年度 修士論文

ノイズに頑健な協調距離計量学習

20M14925 松井 諒生

指導教員 中田 和秀 教授

2022 年 2 月

東京工業大学 工学院 経営工学系

Master's Thesis

Noise Robust Collaborative Metric Learning

Matsui Ryo

Department of Industrial Engineering and Economics

School of Engineering

Tokyo Institute of Technology

Abstract

Recently, there has been a lot of research on recommendation system. Recommender system is a subfield of information retrieval and machine learning, which aim to identify the value of objects and information for each individual. For example, recommendation systems on web services learn latent preferences of users from their behavioral history and display contents which are considered to be preferred by the user. Specifically, recommendation systems on web services have two main requirements: 1) the use of implicit feedback data that does not require users' active actions, and 2) the use of the embedding of users and items. In recent years, a method called Collaborative Metric Learning (CML) has been developed to satisfy the second requirement. However, this method does not deal with the noisy label issues caused by implicit feedback data in the first requirement. In this study, we propose a method to deal with noise in CML comprehensively and effectively. Experimental results show that the proposed method significantly improves the performance on the two requirement compared to the existing methods.

概要

昨今、「推薦システム」の研究が盛んに行われている。推薦システムとは個人個人にとってのモノや情報の価値を特定することを目的とした情報検索・機械学習の一分野である。例えば、web サービスにおいてユーザーの行動履歴から潜在的な嗜好性を学習し、各ユーザーのニーズに合ったコンテンツを「おすすめ」として表示することなどが挙げられる。特に web サービスにおける推薦システムでは、1) ユーザーの嗜好性を学習する際に implicit feedback と呼ばれるユーザーの積極的な行動を必要としないデータを利用する、2) ユーザーの嗜好性を予測・推論する際にユーザーとアイテムの埋め込みベクトルを利用する、という2点が要求される。近年では、後者の要求に応える手法として協調距離計量学習 (Collaborative Metric Learning, CML) と呼ばれる手法が発展してきた。しかし、この手法は前者の要求である implicit feedback に起因するノイズラベル問題に対応していない。そこで本研究では、CML の学習においてノイズを網羅的に扱い、効果的に対処する方法を提案する。実験の結果、提案手法は既存手法と比較して2つの要求に対する性能が大幅に改善することがわかった。

目次

第 1 章	序論	1
1.1	web サービスの要求を満たす推薦システムの特徴	1
1.2	本研究の目的とリサーチクエッション	4
1.3	本論文の構成	5
第 2 章	関連研究	6
2.1	問題の定式化と記号の定義	6
2.2	implicit feedback と埋め込みベクトルによる推薦システム	7
2.3	ノイズラベルを用いた教師あり学習 (信頼学習)	15
第 3 章	提案手法	18
3.1	提案 1: CML に対する二重み付けサンプリングの適用	18
3.2	提案 2: CML に対するクリーニングの適用	21
3.3	理論的な分析	24
第 4 章	数値実験	31
4.1	データセット	31
4.2	比較手法	32
4.3	ハイパーパラメータ	33
4.4	結果	33
第 5 章	結論	38

第 1 章

序論

1.1 web サービスの要求を満たす推薦システムの特徴

動画、音楽、ニュース、本など、あらゆるコンテンツがデジタル化され、web 上で配信されている。インターネットを通じてアクセスできるユーザーにとっては、いつでもどこでも簡単にコンテンツにアクセスでき、利便性がますます向上している。一方で、コンテンツの量が膨大となると、ユーザーは自分の好みに合ったコンテンツを見つけることが難しくなり、偶発的に出会う可能性も低くなってきていると言える。通常、このような膨大なコンテンツ群からそれぞれのユーザーに合った（パーソナライズされた）コンテンツを紹介するには専門家が必要となる。しかし、このような専門家による推薦は対応できるユーザー数および扱うコンテンツ数の観点で限りがある。この問題を解決するため、コンテンツ推薦を自動化する「推薦システム」の研究が盛んに行われている。推薦システムとは一般に、個人個人にとってのモノや情報の価値を特定することを目的とした情報検索・機械学習の一分野である。その中でも特に中心的な web サービスにおける推薦システムはユーザーの属性情報や行動履歴などをもとに、ユーザーそれぞれの好みに合ったアイテム（コンテンツ）を見つけられるようにサポートすることが目的とされる。このようなシステムは経済効果も大きいことから、いまや主要な web サービスにおいては少なからずこのようなシステムが導入されている [1, 2, 3]。

その代表的な例が映画配信プラットフォームにおける推薦システムである。映画配信プラットフォームにおける推薦システムは、視聴者がユーザー、映画がアイテムとみなせ、視聴者の閲覧履歴や「お気に入り」などのフィードバックを用いて、次に視聴するのにおすすめの映画を表示する機能と言える。映画は、一つの作品を消費するために比較的大きな時間、経済的コストを要するため、好みでない映画の閲覧はユーザー体験を大きく毀損する。裏を返せば、好みに合った映画により多く接触させることが大きなユーザー体験の向上につながり、結果として収益を改善させられると言える。実際、米国の大手映画配信プラットフォームは、推薦システムによる収益改善例を数々報告している [4, 5, 6]。

web サービスにおける推薦システムでは主に 2 つの要求がされる。

1 つ目の要求は、ユーザーおよびアイテムの嗜好性や類似性を学習する際に implicit

表 1.1. ユーザーの嗜好に関する 2 種類のデータとその特徴

	explicit feedback データ	implicit feedback データ
データ収集の方法	レビュー・アンケート	閲覧/クリックの履歴
ユーザーの態度	積極的	消極的
データ内のノイズ	少ない	多い
データ収集コスト	大きい	小さい

feedback と呼ばれるデータを利用する点である。今日の web サービスにおいてユーザーの嗜好を表す情報として取得・利用できるデータは explicit feedback と implicit feedback の 2 つに分けられる。explicit feedback は、レビューやアンケートなどのユーザーが積極的に評価を行ったデータであり、5 段階や 10 段階で表される数値のフィードバックを用いることが多い。一方、implicit feedback はそのうちユーザーの能動的な行動を必要としないログデータを表す。例えば、web サイトのクリックや閲覧などの履歴がそれに当てはまる。implicit feedback はユーザーへの調査協力を必要としないため、explicit feedback と比較してデータ収集コストが小さいというメリットがある。しかしながら、クリックや閲覧などの行動が必ずしもユーザーの嗜好を表すとは限らないため、データに含まれるノイズは比較的大きいと言える。このようなトレードオフの関係は、表 1.1 のようにまとめられる。昨今の web サービスではログ技術やビッグデータ処理の技術が発達しているため、ユーザーの能動的な行動を必要としないデータの収集コストがますます下がっている。そのため、implicit feedback のデメリットを考慮したノイズに頑健な推薦システムの構築を目指す研究が盛んに行われている [7, 8, 9, 10, 11, 12]。

主に implicit feedback に起因するノイズは 2 つに分けられる。1 つ目のノイズは、ユーザーがあるアイテムにクリックしたものの、実際にはそのアイテムに興味がなかったという事象を指す**ポジティブノイズ (偽陽性)**である。このようなノイズは、ユーザーのクリック成否がタイトルや説明文の第一印象に左右されやすいことが原因とされている [13, 14]。このようなノイズを無視して学習を行うと、システムは興味のないアイテムをユーザーに推薦してしまうため、ユーザー体験を大きく毀損する。2 つ目のノイズは、ユーザーが自身の嗜好に合ったアイテムを見逃しているという事象を指す**ネガティブノイズ (偽陰性)**である。すなわち、ユーザーが過去にクリックをしていないアイテムの中にも、ユーザーの嗜好に合ったアイテムが含まれるはずだという意味である [11, 15, 16]。このノイズを無視して学習を行うと、単に過去にクリックしたアイテムのみを推薦してしまい、推薦システムの本来の意義を成さない。よって、推薦システムはユーザーがクリックしていないアイテム全てを、ユーザーの嗜好に合わないアイテムとして学習すべきではないと言える。以上より、implicit feedback データを利用するためには、このような 2 種類のノイズについて考慮する必要がある。

web サービスの推薦システムにおける 2 つ目の要求は、ユーザーおよびアイテムの嗜好性や類似性を予測・推論する際に埋め込みベクトルを利用するという点である埋め込みベクトル

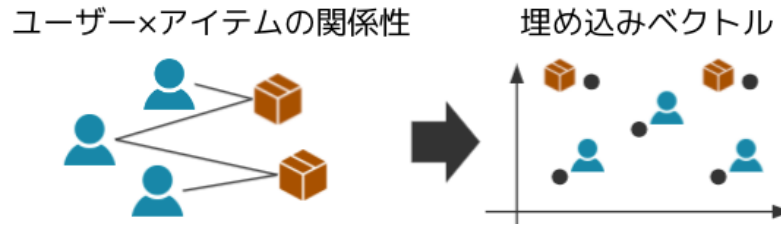


図 1.1. ユーザーアイテムの関係性データと埋め込みベクトルの概観

とは、図 1.1 のように、観測した (ユーザー-アイテム) 間の関係性データをもとに、ユーザーおよびアイテムをそれぞれ 1 つのベクトルとして表現したものである。そして、ユーザーおよびアイテムの潜在的な嗜好や類似性をそのベクトルの類似度によって推定する。この埋め込みベクトルを利用するメリットは主に 2 つある。1 つ目のメリットは低遅延性である。現代 web サービスは大量のユーザーに瞬時にサービスを提供する必要がある。そのため、推薦システムにおいてもユーザーがサービスに到来した際、そのユーザーの嗜好に合うと予想される推薦アイテムを即座に決定する必要がある。埋め込みベクトルを利用しない場合は、そのユーザーの嗜好性を全てのアイテムについて計算し、ソートする必要がある。これは数十億を超えるような膨大な数のアイテムを抱えるサービスの場合現実的でない。一方で埋め込みベクトルを利用した場合、近似近傍探索 [17] などの手法を用いれば、ユーザーベクトルに類似したアイテムベクトルを、ユーザー体験の毀損が起きない時間内で発見することができる。すなわち、全てのアイテムについて嗜好性を計算することを回避した上で、膨大なアイテム群の中から推薦すべきアイテムを発見できる。この結果、先に紹介した映画配信プラットフォームにおける推薦や、検索エンジンにおける web ページの推薦など、膨大なユーザーおよびコンテンツを抱える web サービスにおける推薦が実現する [18]。2 つ目のメリットが (ユーザー-ユーザー) 間および (アイテム-アイテム) 間の類似度も表現できるという点である。ユーザーおよびアイテムを埋め込みベクトルとして表現することによって、ユーザーの潜在的な嗜好に合ったアイテムを捉えるだけでなく、ユーザーと潜在的に類似した別のユーザーや、アイテムと潜在的に類似した別のアイテムを捉えることも可能となる。これにより、顧客セグメントを捉えるマーケティングへの応用や、SNS における友達の推薦、EC サイトにおける関連商品・代替商品の推薦など、幅広い応用が可能となる [2, 19, 20]。

つまり、ここまでをまとめると、web サービスにおける推薦システムの構築にあたって 2 つの要求に応えるには implicit feedback に起因するポジティブノイズおよびネガティブノイズの両者に対処した上で、(ユーザー-アイテム) 間のみならず (ユーザー-ユーザー) 間および (アイテム-アイテム) 間の関係性を高速かつ正確に捉えられる埋め込みベクトルを学習する必要がある。

1.2 本研究の目的とリサーチクエッション

本研究では、前節で示した web サービスにおける 2 つの要求を満たす推薦システムを構築することを目的とする。前節の通り、このような推薦システムを構築するには、以下の 2 つの性質を持つ埋め込みベクトルを獲得できるアルゴリズムを開発すれば良いと言える。

性質 1 implicit feedback に起因する以下 2 つのノイズに頑健である。

- a ポジティブノイズ
- b ネガティブノイズ

性質 2 以下 3 つの全ての関係を精緻に捉えられる。

- a (ユーザー–アイテム) 間
- b (ユーザー–ユーザー) 間
- c (アイテム–アイテム) 間

具体的には、2 点目の性質を満たすために研究されてきた協調距離計量学習 (Collaborative Metric Learning, CML) [21] と呼ばれるアルゴリズムを拡張する。CML は埋め込みベクトルを用いた最も標準的なアルゴリズムである行列分解 [22] が、(ユーザー–ユーザー) 間および (アイテム–アイテム) 間の細かな関係性を表現できていないという指摘 [23, 24] を受けて開発された。行列分解がユーザーとアイテムを表す類似度関数としてベクトルの内積を用いているのと対照的に、CML は類似度関数としてユークリッド距離を採用する。2 章で示すように、距離関数は定義から三角不等式の関係性を満たすため、(ユーザー–アイテム) 間の関係性データのみから学習しても (ユーザー–ユーザー) 間および (アイテム–アイテム) 間の関係性を捉えることができる。このような性質が web サービスの推薦システムの要求に応え、多くの応用や拡張がなされている [25, 26, 27, 28, 29, 30]。

ただし、最も素朴な CML は implicit feedback のノイズに対処していない。Tran ら [31] は CML の学習において部分的にノイズラベル問題へ取り組んだ。CML はユーザー u およびユーザー u とのインタラクションを観測したアイテム i (ポジティブアイテム)、ユーザー u とのインタラクションを観測していないアイテム j (ネガティブアイテム) を加えたトリプレット (u, i, j) を構成単位とした損失関数を持つ。このトリプレットを一様ランダムに抽出し、勾配降下法によるパラメータ更新を繰り返すことによって埋め込みベクトルを学習する。この一様ランダムという点が単純な CML の問題と言える。一様ランダムなサンプリングでは一定確率でノイズなデータを含んでしまうため、学習が効率的でない。この点を解決するため、Tran らはデータの信頼度で重み付けした 2 段階のサンプリングによってノイズの少ないネガティブアイテム i_n を抽出する方法を提案した。

しかしながら、ここまでの研究には依然として 2 つの課題がある。

まず 1 つ目の課題が CML において重み付けのサンプリングが十分でない可能性がある点である。たしかに、重み付けによってノイズのあるネガティブアイテムが抽出する確率が減ら

される。しかし、CML が採用している距離関数の三角不等式の性質上、1 度でもノイズのあるデータがサンプリングされると、ノイズの影響が全体的に広がるため学習に大きな被害を与える。ゆえに、極めてノイズのあるデータをはデータセットから削除する「クリーニング」と呼ばれるアプローチもまた検証すべきである。2 つ目の課題が、ポジティブアイテム i についてのノイズを考慮していない点である。たしかに、ネガティブアイテム j の抽出方法を工夫することでネガティブノイズは軽減できる可能性があるものの、ポジティブノイズもまた重大な悪影響を及ぼしうる。この 2 つの課題を解決するため、本研究では CML の学習の前にデータの信頼度を推定し、ポジティブ・ネガティブの両ノイズに対処した重み付けまたはクリーニングに活用する方法を提案する。ここで信頼度の推定には、ノイズなラベルを含んだ画像分類における最高水準手法である信頼学習 (Confident Learning)[32] を参考にする。本研究の位置付けは図 1.2 のように表される。

既存研究カバー範囲			本研究のカバー範囲		
ノイズの種類 \ アプローチ種類	重み付け	クリーニング	ノイズの種類 \ アプローチ種類	重み付け	クリーニング
Positive Noise	-	-	Positive Noise	✓	✓
Negative Noise	✓	-	Negative Noise	✓	✓

図 1.2. 本研究の位置づけ

そして、本節の冒頭で挙げた推薦システムが満たすべき 2 つの性質に従い、

- 提案手法は既存手法と比較してノイズに頑健に (ユーザー–アイテム) 間の関係性を捉え、より正確な推薦を行うことは可能か。 (性質 1 かつ性質 2-a に対する性能改善)
- 提案手法は既存手法よりもノイズに頑健に (ユーザー–ユーザー) 間、(アイテム–アイテム) 間の関係性を捉えることは可能か。 (性質 1 かつ性質 2-b, 2-c に対する性能改善)

という 2 点を本研究のリサーチクエッションとして設定する。ただし、ここで「ノイズ」は implicit feedback に起因する「ポジティブノイズ」および「ネガティブノイズ」を指す。

1.3 本論文の構成

本論文は本章を含む 5 つの章で構成される。2 章は、問題の定式化と関連研究のアプローチを整理する。さらに、提案手法で参考にするノイズの処理方法についての既存研究も紹介する。3 章では、前節で述べた目的を達成するための提案手法を示し、その有効性について議論する。4 章では、訓練データにノイズを含み、テストデータでノイズのないデータを用いて、2 つのリサーチクエッションに対応した検証を行う。またその結果に基づき、提案手法の実験的な有効性について考察する。そして最後に、5 章において本研究の結論を述べる。なお、本論文は著者らが発表した査読付きの国際会議論文 [33] の内容を精緻化したものとなっている。

第 2 章

関連研究

本章ではまず、2.1 節において web サービスにおける推薦システムを定式化する．次にその定式化に従って、2.2 節にて既存手法のアプローチを紹介する．特に、implicit feedback を利用した埋め込みベクトルをベースとする推薦システムの既存研究について、その発展と現状の問題点を整理する．さらに、2.3 節において教師あり学習の分野で提案されたノイズへのアプローチ方法も紹介し、提案手法を導く足掛かりとする．

2.1 問題の定式化と記号の定義

あるサービスを利用するユーザーの集合を \mathcal{U} ，そのサービスが保有するアイテム集合を \mathcal{I} とする．また、ユーザー数を $N_U = |\mathcal{U}|$ ，アイテム数を $N_I = |\mathcal{I}|$ とする．このサービスにおいて、ある時点でユーザー $u_x \in \mathcal{U}$ が到着したときに、そのユーザーの好みに合うと予想される上位 K 個のアイテム $i_1, i_2, \dots, i_K \in \mathcal{I}$ を推薦することを考える．このユーザーが到着する以前に、ユーザー $u \in \mathcal{U}$ とアイテム $i \in \mathcal{I}$ について、クリックなどのインタラクションが発生していれば $Y_{u,i} = +1$ ，そうでなければ $Y_{u,i} = -1$ と表す．また、この時点で収集した全てのインタラクションの集合を $\mathcal{S} = \{(u, i) | Y_{u,i} = +1\}$ とし、それに含まれないユーザーアイテムペアの集合を $\mathcal{D} = \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$ とする．また、ユーザー u について \mathcal{S} に含まれないアイテム集合を $\mathcal{D}_u = \{i | Y_{u,i} = -1\}$ とする．これらの情報を用いて、推薦を行うために必要なユーザーおよびアイテムに対応した埋め込みベクトル \mathbf{v}_u および \mathbf{v}_i を学習する．そして、ユーザーおよびアイテム同士の関係性を類似度 $\text{Sim}(\mathbf{v}_u, \mathbf{v}_i)$ の値を基に推定することで、未知・未観測な (ユーザー-アイテム) 間の嗜好性や (ユーザー-ユーザー) 間、(アイテム-アイテム) 間の類似性を予測する．ただし、観測したインタラクション $Y_{u,i}$ は必ずしも真のユーザーとアイテムの評価値 (嗜好性) $R_{u,i} \in \{+1, -1\}$ と一致しないことに注意する．真の評価値 $R_{u,i}$ は観測できず、一定のノイズがかかった上で $Y_{u,i}$ が観測されるものとする．そのため、これらのノイズを考慮した上で埋め込みベクトル \mathbf{v}_u および \mathbf{v}_i を学習することが必要となる．

以上の定式化に用いた記号とその定義は表 2.1 のようにまとめられる．

表 2.1. 定式化に用いた記号とその定義

記号	定義
\mathcal{U}	ユーザー集合
\mathcal{I}	アイテム集合
$N_U = \mathcal{U} $	ユーザー数
$N_I = \mathcal{I} $	アイテム数
$u \in \mathcal{U}$	ユーザー
$i \in \mathcal{I}$	アイテム
$K \in \mathbb{N}$	推薦アイテム数
$R_{u,i} \in \{+1, -1\}$	ユーザー u のアイテム i に対する真の評価値
$Y_{u,i} \in \{+1, -1\}$	ユーザー u のアイテム i に対する観測した評価値
\mathbf{v}_u	ユーザー u の埋め込みベクトル
\mathbf{v}_i	アイテム i の埋め込みベクトル
$\mathcal{S} = \{(u, i) Y_{u,i} = +1\}$	ポジティブな評価値を持つユーザーアイテム集合
$\mathcal{D} = \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$	ポジティブな評価値を持たないユーザーアイテム集合
$\mathcal{D}_u = \{i Y_{u,i} = -1\}$	ユーザー u とネガティブな評価値を持つアイテム集合
$\text{Sim}(\cdot, \cdot)$	類似度関数
$d(\cdot, \cdot)$	ユークリッド距離

2.2 implicit feedback と埋め込みベクトルによる推薦システム

本節では、前節で定式化した問題に従っている推薦システムの関連研究を紹介する。特に、implicit feedback データから学習した埋め込みベクトルを用いた最も標準的な手法を基準に、(ユーザー–ユーザー) 間・(アイテム–アイテム) 間の関係性を捉える取り組み、および implicit feedback に起因するノイズへの対応について紹介する。

2.2.1 行列分解 (MF)

最も標準的な推薦システムは行列分解 (Matrix Factorization) をベースとした協調フィルタリングだと言える。この手法は、行にユーザー、列にアイテム、要素に各ユーザーアイテムペアの評価値 $Y_{u,i}$ を持つ行列 $Y \in \mathbb{R}^{N_U \times N_I}$ について、

$$Y \approx U \cdot V$$

のように、2つの低ランク行列 $U \in \mathbb{R}^{N_U \times l}$, $V \in \mathbb{R}^{l \times N_I}$ の積で近似するアルゴリズムである。ただし、 l は近似するランク数を表す。この行列 U, V は、最適化問題

$$U, V = \operatorname{argmin}_{U, V} \|Y - U \cdot V\|_F^2 \quad (2.1)$$

を解くことで達成される。ただし、 $\|\cdot\|_F$ はフロベニウスノルムを表す。これによって求めた U, V を用いて、 $\hat{Y} = U \cdot V$ を再構築することで、潜在的な評価値行列 \hat{Y} を獲得できる。そして、この \hat{Y} をもとに、ユーザーに対して評価の高いアイテムを推薦することが可能となる。

ここで、式 (2.1) は、要素ごとの近似としても表現できる。すなわち、行列 U の行ベクトルを $\mathbf{v}_u (\forall u \in \mathcal{U})$ 、行列 V の列ベクトルを $\mathbf{v}_i (\forall i \in \mathcal{I})$ とみなし、

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} (Y_{u,i} - \hat{Y}_{u,i})^2, \text{ where } \hat{Y}_{u,i} = \mathbf{v}_u^T \mathbf{v}_i$$

とする。ただし Θ は、 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ とし全ての埋め込みベクトル集合を表す。この表記において、ベクトル \mathbf{v}_u およびベクトル \mathbf{v}_i は正にユーザーおよびアイテムの埋め込みベクトルであると言える。また、ユーザーベクトルおよびアイテムベクトルの内積によって関係性・評価値を表していることから、類似度関数を $Sim(u, i) = \mathbf{v}_u^T \mathbf{v}_i$ として考えている手法だとみなせる。

2.2.2 重み付き行列分解 (WMF)

しかし、単純な行列分解は、 $Y_{u,i}$ のノイズを考慮していない。Hu らは $Y_{u,i}$ の信頼性 $c_{u,i}$ で重み付け行列分解 (Weighted MF) を以下のように定式化した [22]。

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} c_{u,i} (Y_{u,i} - \hat{Y}_{u,i})^2, \text{ where } \hat{Y}_{u,i} = \mathbf{v}_u^T \mathbf{v}_i$$

これにより、ノイズの少ないデータ $Y_{u,i}$ を持つユーザーアイテムペアの誤差をより重要視して埋め込みベクトルを獲得することができる。ここで、 $c_{u,i}$ は任意に設定できる。

2.2.3 Bayesian Personalized Ranking (BPR)

さらに、Rendle らはランキング学習による implicit feedback を用いた推薦システムの構築方法を提案した [8]。一般に推薦システムでは、あるユーザーが到来した際、各アイテムとの類似度を計算し、最も類似度の高いアイテムをいくつか推薦する。このとき、類似度の順序が真の (潜在的な) 評価値の順序関係と一致していれば良い推薦と言える。むしろ、類似度は評価値そのものを表現している必要はない。そのため、Rendle らは観測した評価値を近似するのではなく、インタラクションのあった $Y_{u,i} = +1$ であるユーザーアイテムペア (u, i) の類似度がインタラクションのない $Y_{u,j} = -1$ であるユーザーアイテムペア (u, j) の内積よりも相対的に大きくなるよう埋め込みベクトルの学習を行うランキング学習を導入した。特に、Bayesian Personalized Ranking (BPR) は損失

$$\sum_{(u,i) \in \mathcal{S}} \sum_{j \in \mathcal{D}_u} \log(\sigma(\mathbf{v}_u^T \mathbf{v}_j - \mathbf{v}_u^T \mathbf{v}_i)) + \lambda \left(\sum_{u \in \mathcal{U}} \|\mathbf{v}_u\|^2 + \sum_{i \in \mathcal{I}} \|\mathbf{v}_i\|^2 \right)$$

を最小化することで埋め込みベクトルを学習する。ただし、 σ はシグモイド関数 $\sigma(x) = \frac{1}{1 + \exp(-x)}$ を表し、 λ は正則化の強さを表す。ここで、 $\log(\sigma(\cdot))$ が単調増加であるため、ユー

ザー u に対してインタラクションのあったアイテム i との内積を大きく、インタラクションのないアイテム j との内積を小さくすることができる。これによって、推薦システムそのものの目的に対して直接的な学習ができるため、推薦の性能が向上する。

2.2.4 協調距離計量学習 (CML)

上記の行列分解や BPR は埋め込みベクトルの内積によって、ユーザーとアイテムの潜在的な嗜好性を予測している。しかし、Hsieh らはユーザーベクトル同士、アイテムベクトル同士の内積はそれらの関係性の強さを表しておらず、この性質が推薦の精度や埋め込みベクトルの解釈性・拡張性に大きな影響を与えていることを指摘した [21]。

この問題を解決するために同研究で提案された手法が、協調距離計量学習 (Collaborative Metric Learning, CML) である。CML は損失関数を

$$\mathcal{L}^{\text{cml}}(\Theta) = \sum_{(u,i) \in S} \sum_{j \in \mathcal{D}_u} [1 + d(\mathbf{v}_u, \mathbf{v}_i)^2 - d(\mathbf{v}_u, \mathbf{v}_j)^2]_+ \quad (2.2)$$

のように設定し、

$$\min_{\Theta} \mathcal{L}^{\text{cml}}(\Theta), \quad \text{s.t. } \|\mathbf{v}\|^2 \leq 1, \forall \mathbf{v} \in \Theta$$

を解くことによって埋め込みベクトルを学習する。ただし、 $[\cdot]_+$ は関数 $[x]_+ = \min\{0, x\}$ を表す。また、 d はユークリッド距離を表し、これが CML における類似度関数となる。つまり、CML はインタラクションのあった (ユーザー–アイテム) 間の距離を近づけ、インタラクションのない (ユーザー–アイテム) 間の距離を遠ざけるような目的関数を持つ。この類似度関数に距離関数を用いる点が BPR との大きな相違点である。距離関数の三角不等式の性質を用いると、

$$d(u_1, i_1) + d(u_1, i_2) \geq d(i_1, i_2)$$

という関係が得られる。これは、 (u_1, i_1) 間および (u_1, i_2) 間の距離を小さくすることは、 (i_1, i_2) 間の距離の上界を小さくすることを意味する。つまり、同じユーザーとインタラクションしている別のアイテムを類似していると学習できる。同様に、

$$d(u_1, i_1) + d(u_2, i_1) \geq d(u_1, u_2)$$

も得られるため、 (u_1, i_1) 間および (u_2, i_1) 間の距離を小さくすることが、 (u_1, u_2) 間の距離の上界を小さくすることに繋がる。すなわち、同じアイテムとインタラクションしている別のユーザーを類似していると学習できる。さらに、三角不等式を繰り返し用いて、

$$d(u_1, i_1) + d(u_1, i_2) + d(u_2, i_1) \geq d(i_1, i_2) + d(u_2, i_1) \geq d(u_2, i_2)$$

という関係も得られる。これは、図 2.1 のように (u_1, i_1) 間および (u_1, i_2) 間、 (u_2, i_1) 間の距離を小さくすることは、 (u_2, i_2) 間の距離も小さくすることに繋がることを意味する。よって

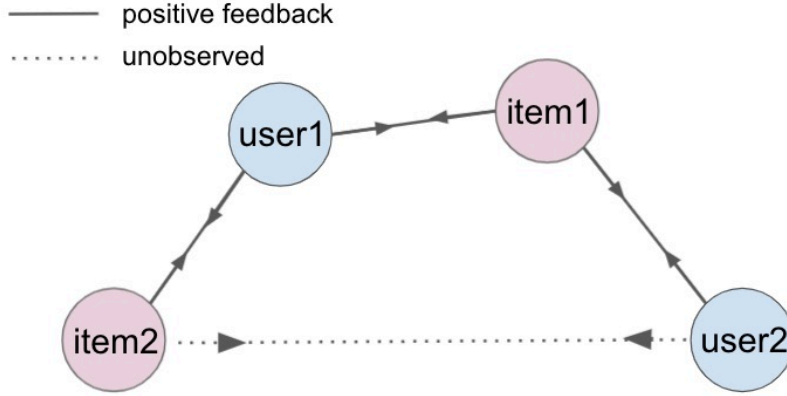


図 2.1. CML における三角不等式の役割を示したイラスト

インタラクションを観測していないユーザーアイテムペアの関係性でも、潜在的にインタラクションが起きやすいユーザーアイテムペアの距離を近づける機構があることが確認できる。

ただし、膨大な組み合わせとなる \mathcal{S}, \mathcal{D} に対して \mathcal{L}^{cml} の大域的最適解を求めることは現実的に難しい。そのため、CML などのトリプレット単位のランキング学習ではミニバッチと呼ばれる部分的なデータをサンプリングし、そのミニバッチにおける損失を最小化する方向にパラメータ更新を繰り返すことで実用的な局所最適解を得る。つまり、初期化された埋め込みベクトルに対して、1) ミニバッチのサンプリング、2) 損失の計算、3) パラメータの更新、を損失が収束するまで繰り返すことで CML が実行できる。(このような操作を確率的勾配降下法と呼ぶ。)

ここで、ミニバッチのサンプリングはポジティブサンプリングとネガティブサンプリングの 2 つに分かれている。まずポジティブサンプリングでは、ポジティブなフィードバック $Y_{u,i} = +1$ を持つユーザーアイテムペア (u, i) を \mathcal{S} から決められた個数ランダムに抽出する。そのサンプルを $\mathcal{B} \subset \mathcal{S}$ と表し、これをポジティブサンプルと呼ぶ。つぎに、 \mathcal{B} に含まれる任意のユーザーアイテムペア (u, i) について、ユーザー u とポジティブなフィードバックをもたない $Y_{u,j} = -1$ なるアイテム j を \mathcal{D}_u から決められた個数ランダムに抽出する。そのアイテムを $\mathcal{N}_{u,i} \subset \mathcal{D}_u$ と表し、これをネガティブサンプルと呼ぶ。ただし、特に言及しない限り両者のサンプリングは重複ありのランダム抽出によって実行される。

また、サンプリングとパラメータ更新のイテレーションによる局所最適解の求解に合わせ、ミニバッチの損失関数も修正される。具体的には、ポジティブサンプリングおよびネガティブサンプリングを通して得られた $\mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}}$ を用いて定義される

$$\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}}) = \sum_{(u,i) \in \mathcal{B}} [1 + d^2(\mathbf{v}_u, \mathbf{v}_i) - \min_{j \in \mathcal{N}_{u,i}} d^2(\mathbf{v}_u, \mathbf{v}_j)]_+ \quad (2.3)$$

を用いて、埋め込みベクトルを更新する。式 (2.2) との大きな相違点は、各ユーザー u についてポジティブなフィードバックを持たないアイテム j に関する損失についてである。式 (2.2) ではユーザー u に対して \mathcal{D}_u に含まれる全てのアイテムについての損失を総和している一方、

式 (2.3) ではユーザー u に対して最近傍に位置するネガティブアイテム j に対する損失のみを考慮する。これは、ユーザーに対して遠くに位置したネガティブアイテムをさらに遠ざけることよりも、より近くに位置したネガティブアイテムを遠ざけることを重視した方がより良い解が得られるという直感を利用した形式である。

これらの一連の操作を Algorithm1 に示す。

2.2.5 重み付きネガティブサンプリング (1stage CML)

BPR や CML はインタラクションのあったユーザーアイテムペア (u, i) にインタラクションのないアイテム j を加えたトリプレット (u, i, j) 単位のランキング学習を行うことで、嗜好性の順序関係のみを学習することにフォーカスした。しかしながら、これらの手法はあくまで $Y_{u,i} \neq R_{u,i}$ であることを無視している。よって、 $Y_{u,i}$ のノイズを考慮することが求められる。

最も簡単なノイズへの対応は、アイテムの人気度を原因としたネガティブノイズを考慮する重み付きネガティブサンプリングである。これはデータセット内の登場回数でネガティブサンプル $\mathcal{N}_{u,i}$ への抽出確率に重みづけする手法である。この手法は、Word2Vec などトリプレットをベースとした表現学習のタスクで有効であることが示されてきている [34][35]。つまり、 $f(j)$ をデータセット内の登場回数としたとき、ネガティブサンプリングにおけるアイテム j の抽出確率 $\Pr(j)$ を以下のように設定する。

$$\Pr(j) = \frac{f(j)^\beta}{\sum_{j'} f(j')^\beta} \quad (2.4)$$

ただし、任意に設定するパラメータ $\beta \geq 0$ はその重み付けの大きさを表し、 $\beta = 0$ のとき一様サンプリングとなる。これは、 $P(R_{u,i} = -1 | Y_{u,i} = -1)$ がアイテムの人気度にのみ依存していると考えている方法と解釈できる。すなわち、人気度が高いにもかかわらずインタラクションが起きていないユーザーとアイテムは潜在的にもインタラクションする可能性が低いという知識を利用したと言える。これらの一連の操作を Algorithm2 に示す。また、Algorithm1 との主な相違点は下線で示す。

2.2.6 2 段階ネガティブサンプリング (2stage CML)

また、Tran らは、重み付きネガティブサンプリングに対してもう 1 段階加えたネガティブサンプリング戦略を提案した [31]。具体的には以下の 2 段階である。

1. 重み付きネガティブサンプリングでネガティブサンプルの候補 $\mathcal{C} \subset \mathcal{I}$ を選択
2. 以下で定義される確率によって、各ユーザーアイテムペア u, i について、候補の中からさらに情報量の多いサンプル $\mathcal{N}_{u,i} \subset \mathcal{C}$ を選択

$$\Pr(j | \mathbf{v}_i) \propto \begin{cases} \frac{1}{p(\mathbf{v}_i^T \mathbf{v}_j = s)}, & 0 \leq s \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.5)$$

$$p(\mathbf{p}_1^T \mathbf{p}_2 = s) = \begin{cases} \frac{(1-s^2)^{\frac{d-1}{2}-1}}{\text{Beta}(\frac{d-1}{2}, \frac{1}{2})} & \text{if } -1 \leq s \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

これらの一連の操作を Algorithm3 に示す. また, 同様に Algorithm1 との主な相違点は下線で示す.

2.2.7 既存手法のまとめと問題点

ここまで述べた既存手法について, その性質を整理すると, 表 2.2 のようにまとめられる.

表 2.2. 各手法が持つ性質の対応表および理想的な手法の性質.

手法	予測対象	類似度関数	ポジティブノイズ への対応	ネガティブノイズ への対応
MF	評価値	内積	なし	なし
WMF	評価値	内積	重み付け	重み付け
BPR	ランキング	内積	なし	なし
CML	ランキング	距離	なし	なし
1stage CML	ランキング	距離	重み付け	なし
2stage CML	ランキング	距離	重み付け	なし
理想的な手法	ランキング	距離	適切な処理	適切な処理

ただし, 類似度関数については,

- 内積: (ユーザー–ユーザー) 間・(アイテム–アイテム) 間の関係を捉えにくい.
- 距離: (ユーザー–ユーザー) 間・(アイテム–アイテム) 間の関係を捉えやすい.

ただしランキング学習においてのみ可能.

という性質があることに注意する. この表より, 1 章でも示した 2 つの要件

1. implicit feedback に起因するノイズへ頑健
2. (ユーザー–ユーザー) 間・(アイテム–アイテム) 間の関係も捉える

を満たす理想的な手法を開発するには, CML を拡張し, 2 つのノイズへ対応していくことが求められる. そこで次節では, ノイズラベルを持つ場合の学習として教師あり学習の分野で提案された手法を説明し, 提案手法を導くための前提とする.

Algorithm 1 協調計量距離学習 (Collaborative Metric Learning, CML)

Input: インタラクションデータ \mathcal{S} ; ユーザー集合 \mathcal{U} ; アイテム集合 \mathcal{I} ; 潜在次元数 l ;
 ポジティブサンプル数 M_+ ; ネガティブサンプル数 M_- ; イテレーション回数 T .

Output: 学習した埋め込みベクトル $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^l \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$

$\mathcal{D} \leftarrow \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$

$\mathcal{D}_u \leftarrow \{i | (u, i) \in \mathcal{D}\} \quad \forall u \in \mathcal{U}$

埋め込みベクトル集合 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ を初期化

for $i = 1, \dots, T$ **do**

$\mathcal{B} \leftarrow$ 一様ランダムに \mathcal{S} から M_+ 個のユーザーアイテムペアを抽出

for $(u, i) \in \mathcal{B}$ **do**

$\mathcal{N}_{u,i} \leftarrow \mathcal{D}_u$ から一様ランダムに M_- 個のネガティブアイテムを抽出

end for

 式 (2.3) に従って損失 $\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}})$ を計算

 損失の勾配 $\frac{\partial \mathcal{L}^{\text{batch}}}{\partial \Theta}$ をもとに Adam[36] を用いて埋め込みベクトル Θ を更新

 任意の埋め込みベクトル $\mathbf{v} \in \Theta$ を $\mathbf{v} := \frac{\mathbf{v}}{\max(\|\mathbf{v}\|, 1)}$ の様に正規化

end for

return \mathbf{v}_*

Algorithm 2 重み付きネガティブサンプリングを用いた協調計量距離学習 (1stage CML)

Input: インタラクションデータ \mathcal{S} ; ユーザー集合 \mathcal{U} ; アイテム集合 \mathcal{I} ; 潜在次元数 l ;

ポジティブサンプル数 M_+ ; ネガティブサンプル数 M_- ; イテレーション回数 T ;

重み付き強度 β

Output: 学習した埋め込みベクトル $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^l \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$

$\mathcal{D} \leftarrow \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$

$\mathcal{D}_u \leftarrow \{i | (u, i) \in \mathcal{D}\} \quad \forall u \in \mathcal{U}$

$\mathcal{S}_i \leftarrow \{u | (u, i) \in \mathcal{S}\} \quad \forall i \in \mathcal{I}$

$f(i) \leftarrow |\mathcal{S}_i| \quad \forall i \in \mathcal{I}$

埋め込みベクトル集合 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ を初期化

for $i = 1, \dots, T$ **do**

$\mathcal{B} \leftarrow$ 一様ランダムに \mathcal{S} から M_+ 個のユーザーアイテムペアを抽出

for $(u, i) \in \mathcal{B}$ **do**

$\mathcal{N}_{u,i} \leftarrow \mathcal{D}_u$ から式 (2.4) に従って M_- 個のネガティブアイテムを抽出

end for

 式 (2.3) に従って損失 $\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}})$ を計算

 損失の勾配 $\frac{\partial \mathcal{L}^{\text{batch}}}{\partial \Theta}$ をもとに Adam[36] を用いて埋め込みベクトル集合 Θ を更新

 任意の埋め込みベクトル $\mathbf{v} \in \Theta$ を $\mathbf{v} := \frac{\mathbf{v}}{\max(\|\mathbf{v}\|, 1)}$ の様に正規化

end for

return \mathbf{v}_*

Algorithm 3 2段階ネガティブサンプリングを用いた協調計量距離学習 (2stage CML)

Input: インタラクションデータ \mathcal{S} ; ユーザー集合 \mathcal{U} ; アイテム集合 \mathcal{I} ; 潜在次元数 l ;
 ポジティブサンプル数 M_+ ; ネガティブサンプル数 M_- ; イテレーション回数 T ;
 重み付き強度 β ; 1段階目候補数 M_C ;

Output: 学習した埋め込みベクトル $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^l \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$

$\mathcal{D} \leftarrow \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$
 $\mathcal{D}_u \leftarrow \{i | (u, i) \in \mathcal{D}\} \quad \forall u \in \mathcal{U}$
 $\mathcal{S}_i \leftarrow \{u | (u, i) \in \mathcal{S}\} \quad \forall i \in \mathcal{I}$
 $f(i) \leftarrow |\mathcal{S}_i| \quad \forall i \in \mathcal{I}$
 埋め込みベクトル集合 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ を初期化

for $i = 1, \dots, T$ **do**
 $\mathcal{B} \leftarrow$ 一様ランダムに \mathcal{S} から M_+ 個のユーザーアイテムペアを抽出
 $\mathcal{C} \leftarrow$ 式 (2.4) に従って \mathcal{I} から M_C 個のアイテム候補を抽出
 for $(u, i) \in \mathcal{B}$ **do**
 $\mathcal{N}_{u,i} \leftarrow \mathcal{C} \cap \mathcal{D}_u$ から式 (2.5) に従って M_- 個のネガティブアイテムを抽出
 end for
 式 (2.3) に従って損失 $\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}})$ を計算
 損失の勾配 $\frac{\partial \mathcal{L}^{\text{batch}}}{\partial \Theta}$ をもとに Adam[36] を用いて埋め込みベクトル集合 Θ を更新
 任意の埋め込みベクトル $\mathbf{v} \in \Theta$ を $\mathbf{v} := \frac{\mathbf{v}}{\max(\|\mathbf{v}\|, 1)}$ の様に正規化
end for
return \mathbf{v}_*

2.3 ノイズラベルを用いた教師あり学習 (信頼学習)

一般に, ある入力 $\mathbf{x} \in \mathbb{R}^d$ とそのラベル $y \in \{1, 2, \dots, M\}$ からなるデータセット $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ を用いて $y \approx f(\mathbf{x})$ なる分類モデル f を学習し, 未観測のデータ \mathbf{x}_{new} に付与されるラベル y_{new} を予測するタスクを教師あり学習における多クラス分類と呼ぶ. その代表的な例が画像分類である. 例えば, 犬の画像と猫の画像のベクトル表現を大量に用いて, 新たに入力される画像が「犬」ラベルか「猫」ラベルかを予測する. 近年では深層学習による手法を中心に世界中で研究され, 一定の成果を挙げている.

しかしながら, 実務上の課題としてノイズラベル問題が指摘されている. 教師あり学習は, 多くのラベル付きデータを要する. 先に挙げた画像分類の例では想像に難くないが, タスクによっては人が1データずつ目で判断しなくてはならないなど, このラベルを収集するコストが大きい場合がある. このような場合, データ収集コスト削減のため, クラウドソーシングなど

によって不特定多数の人にデータ作成を依頼するケースもあり、そのデータ作成の質が低いケースも散見される。また、専門家でもそもそもラベルを与えることが難しい場合もある。例えば医療分野において、CTR 画像から病気を発見する応用が提案されているが、医者でも全ての画像に完全に正しく病気を割り当てられないと言われている。これらのような状況において教師あり学習を行う際に注意すべき点がノイズラベル問題である。すなわち、それぞれのデータに与えられたラベルが必ずしも正解ではないということを無視すべきでないという問題設定である。

このようなノイズラベル問題への対策は、実応用が進むにつれて近年研究がされている。特に、推薦システムにも応用が可能と考えられる手法は大きく2つ存在する。

1つは、前節でも紹介した重み付けをベースとした手法である。この方法は、事前もしくは学習中にデータの信頼度を推定し、損失関数もしくは確率的勾配降下法におけるミニバッチのサンプリング確率をその信頼度をもとに重み付けするという手順をとる。これにより、ノイズでないといみなせるデータの誤差を重視して学習することができる。これは推薦システムにも比較的容易に実装可能であるため、前節で紹介した重み付き行列分解をはじめとするいくつかの既存研究が存在する。

2つ目が、クリーニングをベースとした手法である。この方法は、事前にデータの信頼度を推定し、信頼できないデータを除去した上で通常の教師あり学習を行う。特に、Northcutt らは教師あり学習の分類予測タスクにおける汎用的な方法として、信頼学習 (Confident learning) と呼ばれる手法を提案した [32]。以下でその概要を示す。

信頼学習 (Confident Learning)

特徴量 $\mathbf{x} \in \mathbb{R}^d$ およびノイズを含むラベル $\tilde{y} \in \{1, 2, \dots, M\}$ からなるデータセット $\tilde{D} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^N$ を用いて、新たな入力 \mathbf{x}_{new} の各ラベル $j \in \{1, 2, \dots, M\}$ への所属確率 $p(y_{\text{new}} = j | \mathbf{x}_{\text{new}})$ を予測するタスクを考える。すなわち、任意のパラメータ Θ でパラメータ化された関数 $f(\mathbf{x} | \Theta), f: \mathbb{R}^d \rightarrow \mathbb{R}^M$ を用いて、

$$p(y_{\text{new}} = j | \mathbf{x}_{\text{new}}) \approx \hat{p}(y_{\text{new}} = j | \mathbf{x}_{\text{new}}) = \text{Softmax}_j(f(\mathbf{x}_{\text{new}} | \Theta)), \quad j = 1, \dots, M$$

となるよう Θ を最適化する。ただし、 Softmax_j は、入力ベクトル $\mathbf{z} \in \mathbb{R}^M$ の各要素 $z_j, (j = 1, \dots, M)$ を用いて、 j 番目のラベルへの所属確率 $p_j = \frac{\exp(z_j)}{\sum_{j'=1}^M \exp(z_{j'})}$ に変換する関数である。ただし、必ずしも \tilde{y}_i が真のラベル y とは異なることに注意する。信頼学習は、このような設定において、 Θ や f の形状および最適化の手順に依らず、以下の手順によって実行される。

1. まず単純にノイズデータ \tilde{D} を用いてモデル $\tilde{f}(\mathbf{x} | \tilde{\Theta})$ を学習する。
2. 各データ $i = 1, \dots, N$ および各ラベル $j = 1, \dots, M$ について、モデル \tilde{f} を用いて以下で定義されるノイズ率 ρ_{j, \mathbf{x}_i} を計算する。

$$\rho_{j, \mathbf{x}_i} = \hat{p}(y = j | \mathbf{x}_i) - \hat{p}(y = \tilde{y}_i | \mathbf{x}_i) \quad (2.6)$$

$$= \text{Softmax}_j(\tilde{f}(\mathbf{x}_i | \tilde{\Theta})) - \text{Softmax}_{\tilde{y}_i}(\tilde{f}(\mathbf{x}_i | \tilde{\Theta})) \quad (2.7)$$

3. 各ラベル $j = 1, \dots, M$ について, ρ_{j, \mathbf{x}_i} が大きいデータ (\mathbf{x}_i, y_i) を \mathcal{D} から除外し, データセット \mathcal{D}' を作成する.
4. \mathcal{D}' を用いてモデル f を学習する.

これらの手順を要約すると, ノイズを含むデータで学習・予測した各ラベルの所属確率をもとにノイズ率を算出しデータをクリーニングする方法と言える. 従ってこの方法は, ノイズラベルを用いて各ラベルへの所属確率が推定可能であれば適用が可能であると言える. よって, 推薦システムでもノイズラベル $Y_{u,i}$ の所属確率 $P(Y_{u,i} = +1), P(Y_{u,i} = -1)$ を推定することで, これと類似した方法を適用することができると考えられる.

以上のように, 推薦システムに適用可能と考えられるノイズへの対応は, 1) 重み付け, 2) クリーニングと2種類存在するが, これらには情報損失とノイズへの脆弱性の間でトレードオフの関係がある. 重み付けは, データを切り捨てないため情報損失が少ないが, 一定確率でノイズが含まれてしまう. 一方で, クリーニングは, データ削除割合を適切に定めれば確実にノイズのあるデータが削除され, ノイズの影響が軽減できるが, ノイズのないデータも失われてしまい, 学習の効率性を損なう可能性がある. これらは, 状況や問題に応じて取るべき手法が異なると言える.

第 3 章

提案手法

本章では、1 章で示した目的を達成するための推薦システムを提案する．特に、2 章で列挙した既存手法およびその問題点を解決するため、既存研究のひとつである協調計量距離学習 (CML) の拡張を考える．具体的には、教師あり学習の分野で開発されたノイズへの対応方法を参考に、ポジティブノイズおよびネガティブノイズの両者に対して処理を施す．また、その処理方法として重み付けおよびクリーニングの手法を提案し、どちらが性能が良いかを後述する数値実験において検証する．

したがって、本章では 3.1 節においてまず重み付けの手法について先に提案し、次に 3.1 節においてクリーニングの手法を続いて提案する．さらに、3.3 節にてそれらの理論的な性質についても議論し、その有効性を分析する．

3.1 提案 1：CML に対する双重み付けサンプリングの適用

本節では、ポジティブサンプリングおよびネガティブサンプリングの両者に対して重み付けを行った CML の学習方法を提案する．

ここで、ポジティブノイズを考慮して、ポジティブサンプリングを重み付けする目的は、潜在的にユーザー u がそのアイテム i を好んでいることを表す $R_{u,i} = +1$ である可能性の高いユーザーアイテムペアをより重視して学習することである．同様に、ネガティブノイズを考慮してネガティブサンプリングを重み付けする目的は、潜在的にユーザー u がそのアイテム i を好んでいないことを表す $R_{u,i} = -1$ である可能性の高いユーザーアイテムペアをより重視して学習することである．しかしながら、この $R_{u,i}$ は手に入らず、ノイズのかかった $Y_{u,i}$ のみが手に入る．したがって、 $Y_{u,i}$ のみのデータから、 $R_{u,i} = +1$ である確率 $P(R_{u,i} = +1)$ および $R_{u,i} = -1$ である可能性 $P(R_{u,i} = -1)$ を推定する必要がある．ここについて既存研究では、ネガティブサンプリングのみについてフォーカスし、 $P(R_{u,i} = -1)$ をアイテム i の人気度に比例すると仮定した．つまり、 $f(j) = |\{u | Y_{u,j} = +1\}|$ として $f(j)$ で重み付けする．しかしながら、これはポジティブラベル $Y_{u,j} = +1$ にノイズがないことを暗に仮定しており、ノイズのある状況においては機能しないことが予想される．

そこで、本研究では $P(R_{u,i} = \pm 1)$ の代替として $Y_{u,i} = +1$ である確率 $P(Y_{u,i} = \pm 1)$ を事前に推定して用いることを提案する。この代替の妥当性は続く 3.3.1 節において議論する。

3.1.1 信頼度の推定

与えられたラベル $Y_{u,i}$ を用いて $P(Y_{u,i} = +1)$ を推定する。これは、任意のパラメータ Φ でパラメータ化された任意のモデル $g: \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ を用いて、

$$P(Y_{u,i} = +1) \approx \hat{P}(Y_{u,i} = +1) = \sigma(g(u, i; \Phi)) \quad (3.1)$$

として Φ を学習することで可能となる。ただし、 σ はシグモイド関数を表す。このとき、 $P(Y_{u,i} = 1)$ が独立に $\sigma(g(u, i; \Phi))$ を母数とするベルヌーイ分布に従うとすると、負の対数尤度関数

$$L(\tilde{\Phi}) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} -Y_{u,i} \log \sigma(g(u, i; \Phi)) - (1 - Y_{u,i}) \log \sigma(-g(u, i; \Phi)) \quad (3.2)$$

を最小化する Φ を求める問題として考えられる。ここで、具体的な g や Φ には、2 章で紹介した行列分解の拡張であるロジスティック行列分解 [37] や Factorization Machines [38] などの一般的な推薦システムに用いられるモデルを採用する。ただし、モデル g は 1 章で示した要件を満たす必要はなく、より条件の緩いモデルを採用できる。例えばロジスティック行列分解では、ユーザーおよびアイテムの埋め込みベクトルおよびバイアス項の集合 $\tilde{\Theta} = (\{\mathbf{v}_c, b_c\}_{c \in \mathcal{U} \cup \mathcal{I}})$ に対して、モデル \tilde{f} を

$$\tilde{f}(u, i; \tilde{\Theta}) = \mathbf{v}_u^T \mathbf{v}_i + b_u + b_i, \quad \tilde{\Theta} = (\{\mathbf{v}_c, b_c\}_{c \in \mathcal{U} \cup \mathcal{I}})$$

と定義する。この \tilde{f} を式 (3.1) における g 、 $\tilde{\Theta}$ を式 (3.1) における Φ とみなせば、式 (3.2) で表される L を明示的に定式化できる。そして L を最小にするパラメータ $\tilde{\Theta}$ を非線形連続最適化アルゴリズムの Adam [36] によって求めれば $\hat{P}(Y_{u,i} = +1)$ を推定する $\tilde{f}(u, i; \tilde{\Theta})$ を構築できる。また当然これを用いれば $P(Y_{u,i} = -1)$ についても $\hat{P}(Y_{u,i} = -1) = 1 - \hat{P}(Y_{u,i} = +1)$ として推定可能である。

3.1.2 二重み付けサンプリング

$P(R_{u,i} = \pm 1)$ の代わりに推定した $P(Y_{u,i} = \pm 1)$ を用いてポジティブサンプリングおよびネガティブサンプリングの両者に重み付けする方法を示す。まず、ポジティブサンプリングは、ポジティブフィードバック集合 \mathcal{S} からミニバッチとしてユーザーアイテムペア (u, i) を抽出する確率 $\Pr(u, i)$ を、 $P(Y_{u,i} = +1)$ を用いて

$$\Pr(u, i) = \frac{P(Y_{u,i} = +1)}{\sum_{(u, i') \in \mathcal{S}} P(Y_{u, i'} = +1)} \quad (3.3)$$

として設定する。つまり、 $P(Y_{u,i} = +1)$ に比例した重み付けをする。これにより、嗜好の合ったユーザーアイテムペアをよりポジティブな関係として重視した学習を行うことができる。

次に、ネガティブサンプリングは、ポジティブサンプリングで抽出した各ユーザーアイテムペア (u, i) に対して、アイテム j を抽出する確率 $\Pr_u(j)$ を、 $P(Y_{u,j} = -1)$ を用いて

$$\Pr_u(j) = \frac{P(Y_{u,i} = -1)}{\sum_{j' \in \mathcal{D}_u} P(Y_{u,j'} = -1)} \quad (3.4)$$

として設定する．これも同様に、 $P(Y_{u,i} = -1)$ に比例した重み付けと言える．これにより、嗜好の合っていないユーザーアイテムペアをよりネガティブな関係として重視した学習を行うことができる．以上の一連の操作を Algorithm 4 に示す．また、Algorithm1(CML) との主な相違点は下線で示す．

Algorithm 4 Dual Weighted Collaborative Metric Learning

Input: インタラクシオンデータ \mathcal{S} ; ユーザー集合 \mathcal{U} ; アイテム集合 \mathcal{I} ; 潜在次元数 l ;
 ポジティブサンプル数 M_+ ; ネガティブサンプル数 M_- ; イテレーション回数 T ;
 インタラクシオン確率推定モデル \tilde{f}

Output: 学習した埋め込みベクトル $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^l \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$

$\mathcal{D} \leftarrow \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$

インタラクシオン確率推定モデル \tilde{f} のパラメータ Θ を \mathcal{S} および \mathcal{D} を用いて学習

$P(Y_{u,i} = +1) \leftarrow \sigma(\tilde{f}(u, i; \Theta)) \quad \forall (u, i) \in \mathcal{S}$

$P(Y_{u,i} = -1) \leftarrow 1 - \sigma(\tilde{f}(u, i; \Theta)) \quad \forall (u, i) \in \mathcal{D}$

埋め込みベクトル集合 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ を初期化

for $i = 1, \dots, T$ **do**

$\mathcal{B} \leftarrow$ 式 (3.3) に従って \mathcal{S} から M_+ 個のユーザーアイテムペアを抽出

for $(u, i) \in \mathcal{B}$ **do**

$\mathcal{N}_{u,i} \leftarrow$ 式 (3.4) に従って M_- 個のネガティブアイテムを抽出

end for

 式 (2.3) に従って損失 $\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}})$ を計算

 損失の勾配 $\frac{\partial \mathcal{L}^{\text{batch}}}{\partial \Theta}$ をもとに Adam[36] を用いて埋め込みベクトル Θ を更新

 任意の埋め込みベクトル $\mathbf{v} \in \Theta$ を $\mathbf{v} := \frac{\mathbf{v}}{\max(\|\mathbf{v}\|, 1)}$ の様に正規化

end for

return \mathbf{v}_*

3.2 提案 2：CML に対するクリーニングの適用

次に、ポジティブサンプリングおよびネガティブサンプリングの両者に対してクリーニングの処理を施した CML の学習方法を提案する．特に，教師あり学習における信頼学習では入力する特徴量 \mathbf{x} およびノイズのあるラベル \tilde{y} について， $\tilde{y} \neq j$ なるラベルに対するノイズ率 ρ_{j,\mathbf{x}_i} を，

$$\rho_{j,\mathbf{x}} = \tilde{p}(y = j|\mathbf{x}) - \tilde{p}(y = \tilde{y}|\mathbf{x}) \quad (3.5)$$

と定義した．そして，このノイズ率が大きいサンプルをデータから削除することを考えた．本節ではこの手法を CML に応用する．

3.2.1 ノイズ率の推定

推薦システムでは式 (3.5) において，入力 \mathbf{x} をユーザーアイテムペア (u, i) ，ノイズのあるラベル \tilde{y} を観測したインタラクション $Y_{u,i}$ ，ラベル j が取りうる集合を $\{-1, +1\}$ とみなすことができる．これを用いて，インタラクションを観測したユーザーアイテムペアに関するノイズ率 $\rho_{u,i}^+$ および，インタラクションを観測していないユーザーアイテムペアに関するノイズ率 $\rho_{u,i}^-$ を，

$$\rho_{u,i}^+ = \hat{P}(Y_{u,i} = -1) - \hat{P}(Y_{u,i} = +1) \quad (3.6)$$

$$\rho_{u,i}^- = \hat{P}(Y_{u,i} = +1) - \hat{P}(Y_{u,i} = -1) \quad (3.7)$$

として表すことができる．さらに，

$$\hat{P}(Y_{u,i} = -1) = 1 - \hat{P}(Y_{u,i} = +1)$$

であることを利用すると，

$$\rho_{u,i}^+ = 1 - 2\hat{P}(Y_{u,i} = +1)$$

$$\rho_{u,i}^- = 2\hat{P}(Y_{u,i} = +1) - 1$$

と書き下すことができる．ここで， $\hat{P}(Y_{u,i} = +1)$ はまさに前節で重みに利用した信頼度と一致している．したがって，前節の二重み付けサンプリングと同様な方法によって，任意のユーザーアイテムペア (u, i) について $\hat{P}(Y_{u,i} = +1)$ を $\hat{P}(Y_{u,i} = +1) = \sigma(g(u, i|\Phi))$ として計算可能な $g(u, i|\Phi)$ を構築すればノイズ率 $\rho_{u,i}^+, \rho_{u,i}^-$ も計算できる．よって，この $\rho_{u,i}^+$ が大きいユーザーアイテムペア $(u, i) \in S$ および， $\rho_{u,i}^-$ が大きいユーザーアイテムペア $(u, j) \in D$ を学習対象から除くことで，ノイズのあるラベルが学習に使われなくなり，クリーニングが達成されると言える．

ここで，教師あり学習における信頼学習を推薦システムに拡張した際の対応関係は表 3.1 のように整理することができる．

表 3.1. 信頼学習の推薦システムへの適用時の対応関係

	教師あり学習	推薦システム (提案)
入力	特徴量 \mathbf{x}_i	ユーザーアイテムペア (u, i)
ノイズラベル	クラスラベル \tilde{y}_i	インタラクション $Y_{u,i}$
取りうるラベル	ラベル集合 $\{1, 2, \dots, M\}$	フィードバック $\{-1, +1\}$
推定モデル	$\tilde{f}(\mathbf{x}_i \tilde{\Theta})$	$\tilde{f}(u, i \tilde{\Theta})$

3.2.2 クリーニングと学習方法

次に、前節で推定した $\rho_{u,i}^+, \rho_{u,i}^-$ を用いて、CML でサンプリングの対象とすべきユーザーアイテムペアを決定する。この処理はポジティブおよびネガティブのサンプリングの両者について行う。まずポジティブサンプリングについて、はじめに削除するデータ割合 p を任意に決定する。そして、閾値 t_+ を

$$t_+ = \text{Percentile}(\{\rho_{u,i}^+\}_{(u,i) \in \mathcal{S}}, p)$$

として求める。ただし、 $\text{Percentile}(\mathcal{A}, p)$ はスカラーを要素とする集合 \mathcal{A} の上位 $p\%$ 点を表す。続いて、

$$\mathcal{S}' = \{(u, i) | (u, i) \in \mathcal{S} \wedge \rho_{u,i}^+ \leq t_+\}$$

とすることで、インタラクションを観測していないユーザーアイテムペアの集合についてノイズを除去した集合 $\mathcal{S}' \subset \mathcal{S}$ を得ることができる。一方、ネガティブサンプリングについても同様に、まずは削除するデータ割合 q を任意に決定する。そして、閾値 t_- を

$$t_- = \text{Percentile}(\{\rho_{u,i}^-\}_{(u,i) \in \mathcal{D}}, q)$$

のように求める。続いて、

$$\mathcal{D}' = \{(u, i) | (u, i) \in \mathcal{D} \wedge \rho_{u,i}^- \leq t_-\}$$

とすることで、インタラクションを観測していないユーザーアイテムペアの集合についてノイズの小さい部分集合 $\mathcal{D}' \subset \mathcal{D}$ を得ることができる。また、ユーザー u とインタラクションを観測していないアイテム集合についてノイズの小さい部分集合 \mathcal{D}'_u は、 $\mathcal{D}'_u = \{i | (u, i) \in \mathcal{D}'\}$ と表せる。そして、これらを用いて

- **ポジティブサンプリング:** ノイズの小さい集合 \mathcal{S}' から \mathcal{B} を抽出
- **ネガティブサンプリング:** 任意の $(u, i) \in \mathcal{B}$ について、ノイズの小さい集合 \mathcal{D}'_u から $\mathcal{N}_{u,i}$ を抽出

という手順でサンプリングを行えば、CML の学習においてクリーニングが実行可能となる。これらの一連の手順をアルゴリズム 5 に示す。また、Algorithm1(CML) との主な相違点は下線で示す。

Algorithm 5 Confident Collaborative Metric Learning

Input: インタラクションデータ \mathcal{S} ; ユーザー集合 \mathcal{U} ; アイテム集合 \mathcal{I} ; 潜在次元数 l ;
 ポジティブサンプル数 M_+ ; ネガティブサンプル数 M_- ; イテレーション回数 T ;
 ポジティブ除外割合 p ; ネガティブ除外割合 q ; インタラクション確率推定モデル f

Output: 学習した埋め込みベクトル $\mathbf{v}_u, \mathbf{v}_i \in \mathbb{R}^l \forall u \in \mathcal{U}, \forall i \in \mathcal{I}$

$\mathcal{D} \leftarrow \mathcal{U} \times \mathcal{I} \setminus \mathcal{S}$

インタラクション確率推定モデル \tilde{f} のパラメータ Θ を \mathcal{S} および \mathcal{D} を用いて学習

$\rho_{u,i}^+ \leftarrow 1 - 2\sigma(\tilde{f}(u, i; \Theta)) \quad \forall (u, i) \in \mathcal{S}$

$\rho_{u,i}^- \leftarrow 2\sigma(\tilde{f}(u, i; \Theta)) - 1 \quad \forall (u, i) \in \mathcal{D}$

$t_+ \leftarrow \text{Percentile}(\{\rho_{u,i}^+\}_{(u,i) \in \mathcal{S}}, p)$

$t_- \leftarrow \text{Percentile}(\{\rho_{u,i}^-\}_{(u,i) \in \mathcal{D}}, q)$

$\mathcal{S}' \leftarrow \{(u, i) | (u, i) \in \mathcal{S} \wedge \rho_{u,i}^+ \leq t_+\}$

$\mathcal{D}' \leftarrow \{(u, i) | (u, i) \in \mathcal{D} \wedge \rho_{u,i}^- \leq t_-\}$

$\mathcal{D}'_u \leftarrow \{i | (u, i) \in \mathcal{D}'\}$

埋め込みベクトル集合 $\Theta = \{\mathbf{v}_c\}_{c \in \mathcal{U} \cup \mathcal{I}}$ を初期化

for $i = 1, \dots, T$ **do**

$\mathcal{B} \leftarrow$ 一様ランダムに \mathcal{S}' から M_+ 個のユーザーアイテムペアを抽出

for $(u, i) \in \mathcal{B}$ **do**

$\mathcal{N}_{u,i} \leftarrow$ 一様ランダムに \mathcal{D}'_u から M_- 個のネガティブアイテムを抽出

end for

式 (2.3) に従って損失 $\mathcal{L}^{\text{batch}}(\Theta, \mathcal{B}, \{\mathcal{N}_{u,i}\}_{(u,i) \in \mathcal{B}})$ を計算

損失の勾配 $\frac{\partial \mathcal{L}^{\text{batch}}}{\partial \Theta}$ をもとに Adam[36] を用いて埋め込みベクトル Θ を更新

任意の埋め込みベクトル $\mathbf{v} \in \Theta$ を $\mathbf{v} := \frac{\mathbf{v}}{\max(\|\mathbf{v}\|, 1)}$ の様に正規化

end for

return \mathbf{v}_*

3.3 理論的な分析

本節では、提案手法 1 および提案手法 2 の理論的な性質を示す．まずは $P(R_{u,i} = \pm 1)$ の代替として採用した信頼度 $P(Y_{u,i} = \pm 1)$ の妥当性について議論し、次に 2 つの提案手法の比較、最後に大規模データへの適用可能性を議論する．

3.3.1 信頼度およびノイズ率の妥当性

まずは、 $\hat{P}(R_{u,i} = \pm 1)$ の代替として採用した信頼度 $\hat{P}(Y_{u,i} = \pm 1)$ の妥当性について考える．先述の通り、1 つ目の提案手法である重み付けサンプリングでは、 $\hat{P}(R_{u,i} = \pm 1)$ の重み付けの代替として $\hat{P}(Y_{u,i} = \pm 1)$ を採用した．また、クリーニングでは式 (3.6)、式 (3.7) で表されるノイズ率の大小によってサンプリング対象から除去すべきユーザーアイテムペアを決定した．この定義において $\rho_{u,i}^+$ は $\hat{P}(Y_{u,i} = +1)$ に対して単調減少、 $\rho_{u,i}^-$ は $\hat{P}(Y_{u,i} = +1)$ に対して単調増加であるため、実質的に $\hat{P}(Y_{u,i} = +1)$ が小さいユーザーアイテムペアをポジティブサンプリングから除去し、 $\hat{P}(Y_{u,i} = +1)$ が大きいユーザーアイテムペアをネガティブサンプリングから除去していると捉えられる．したがって、本研究の 2 つの提案手法はどちらも $P(Y_{u,i} = \pm 1)$ を推定し、それがあたかも $P(R_{u,i} = \pm 1)$ を表すかのように考えてノイズ処理に応用する手法だとみなすことができる．よってここでは、 $P(R_{u,i} = \pm 1)$ の代替として $P(Y_{u,i} = \pm 1)$ がどのように機能するかを示す．特に、その誤差 $P(R_{u,i} = \pm 1) - P(Y_{u,i} = \pm 1)$ の上界および下界を示すことで、重み付けやクリーニングへ応用した際の性質を整理する．

これにあたって、まずは潜在的な (真の) ノイズ率を

$$\sigma_{u,i}^+ = P(Y_{u,i} = +1 | R_{u,i} = -1) \geq 0$$

$$\sigma_{u,i}^- = P(Y_{u,i} = -1 | R_{u,i} = +1) \geq 0$$

と表す．これを用いると、

$$-\sigma_{u,i}^- \leq P(R_{u,i} = +1) - P(Y_{u,i} = +1) \leq \sigma_{u,i}^+ \quad (3.8)$$

$$-\sigma_{u,i}^+ \leq P(R_{u,i} = -1) - P(Y_{u,i} = -1) \leq \sigma_{u,i}^- \quad (3.9)$$

を示すことができる．

証明. 式 (3.8) および式 (3.9) を示す．

$P(Y_{u,i} = +1)$ は以下の通り $P(R_{u,i} = +1)$ および $\sigma_{u,i}^+, \sigma_{u,i}^-$ を用いて表すことができる．

$$\begin{aligned} P(Y_{u,i} = +1) &= P(R_{u,i} = +1)P(Y_{u,i} = +1 | R_{u,i} = +1) \\ &\quad + P(R_{u,i} = -1)P(Y_{u,i} = +1 | R_{u,i} = -1) \\ &= P(R_{u,i} = +1)(1 - \sigma_{u,i}^-) + (1 - P(R_{u,i} = +1))\sigma_{u,i}^+ \\ &= P(R_{u,i} = +1)(1 - \sigma_{u,i}^+ - \sigma_{u,i}^-) + \sigma_{u,i}^+ \end{aligned} \quad (3.10)$$

また, $P(Y_{u,i} = -1)$ も同様に,

$$\begin{aligned}
 P(Y_{u,i} = -1) &= P(R_{u,i} = -1)P(Y_{u,i} = -1|R_{u,i} = -1) \\
 &\quad + P(R_{u,i} = +1)P(Y_{u,i} = -1|R_{u,i} = +1) \\
 &= P(R_{u,i} = -1)(1 - \sigma_{u,i}^+) + (1 - P(R_{u,i} = -1))\sigma_{u,i}^- \\
 &= P(R_{u,i} = -1)(1 - \sigma_{u,i}^- - \sigma_{u,i}^+) + \sigma_{u,i}^-
 \end{aligned} \tag{3.11}$$

と表せる. さらに, 式 (3.10) を $P(R_{u,i} = +1)$ について解くことで

$$P(R_{u,i} = +1) = \frac{P(Y_{u,i} = +1) - \sigma_{u,i}^-}{1 - \sigma_{u,i}^- - \sigma_{u,i}^+} \geq P(Y_{u,i} = +1) - \sigma_{u,i}^-$$

が得られ, 式変形をすることで

$$P(R_{u,i} = +1) - P(Y_{u,i} = +1) \geq -\sigma_{u,i}^- \tag{3.12}$$

が得られる. 同様に式 (3.11) も $P(R_{u,i} = -1)$ について解くことで

$$P(R_{u,i} = -1) = \frac{P(Y_{u,i} = -1) - \sigma_{u,i}^+}{1 - \sigma_{u,i}^+ - \sigma_{u,i}^-} \geq P(Y_{u,i} = -1) - \sigma_{u,i}^+$$

が得られ, 式変形することで

$$P(R_{u,i} = -1) \geq P(Y_{u,i} = -1) - \sigma_{u,i}^+ \tag{3.13}$$

が得られる. さらに, 式 (3.13) に

$$\begin{aligned}
 P(R_{u,i} = -1) &= 1 - P(R_{u,i} = +1), \\
 P(Y_{u,i} = -1) &= 1 - P(Y_{u,i} = +1)
 \end{aligned}$$

を代入すると,

$$\begin{aligned}
 \{1 - P(R_{u,i} = +1)\} - \{1 - P(Y_{u,i} = -1)\} &\geq -\sigma_{u,i}^+ \\
 \Leftrightarrow P(R_{u,i} = +1) - P(Y_{u,i} = +1) &\leq \sigma_{u,i}^+
 \end{aligned} \tag{3.14}$$

となり, 同様に式 (3.12) に

$$\begin{aligned}
 P(R_{u,i} = +1) &= 1 - P(R_{u,i} = -1), \\
 P(Y_{u,i} = +1) &= 1 - P(Y_{u,i} = -1)
 \end{aligned}$$

を代入すると,

$$\begin{aligned}
 \{1 - P(R_{u,i} = -1)\} - \{1 - P(Y_{u,i} = +1)\} &\geq -\sigma_{u,i}^- \\
 \Leftrightarrow P(R_{u,i} = -1) - P(Y_{u,i} = -1) &\leq \sigma_{u,i}^-
 \end{aligned} \tag{3.15}$$

となる. よって, 式 (3.12) および式 (3.14) より式 (3.8) が得られ, 式 (3.13) および式 (3.15) より式 (3.9) が得られる. ただし, 等号の成立は $\sigma_{u,i}^+ = \sigma_{u,i}^- = 0$ のときである.

証明終了

□

上記で示した式 (3.8) および式 (3.9) は、 $P(Y_{u,i} = \pm 1)$ の $P(R_{u,i} = \pm 1)$ に対する誤差がノイズ率 $\sigma_{u,i}^+, \sigma_{u,i}^-$ によって上下から抑えられることを表している。これは、ノイズ率が十分小さいとき $P(R_{u,i} = \pm 1)$ と $P(Y_{u,i} = \pm 1)$ の差も小さくなることを意味する。したがって、潜在的なノイズ率が低かったにも関わらず発生してしまったノイズについては $P(R_{u,i} = \pm 1)$ を $P(Y_{u,i} = \pm 1)$ で代替して重み付けやクリーニングを行うことで十分ノイズ除去が可能だと言える。しかしながら、そもそも潜在的なノイズ率が大きいユーザーアイテムペアが存在すると、大きな誤差を生む可能性がある。具体的には、 $\sigma_{u,i}^-$ が大きい場合に $P(Y_{u,i} = +1)$ は $P(R_{u,i} = +1)$ よりも過大評価される可能性が高く、同様に $\sigma_{u,i}^+$ が大きい場合に $P(Y_{u,i} = -1)$ は $P(R_{u,i} = -1)$ よりも過大評価される可能性が高い。つまり、根本的なノイズ発生率が高い、いわば「現れるべくして現れたノイズ」については脆弱な可能性がある。

しかし、潜在的なノイズ率は一般に計測することができない。また、現実世界でのノイズの発生メカニズムについても明らかではないため、シミュレーションを行うのは難しい。そのため、本研究では実世界で収集した複数のデータセットにおいて性能を検証することで、これらの実用性を評価することにする。

3.3.2 2つの提案手法の比較

前節の通り、本研究の2つの提案手法はどちらも信頼度に応じてサンプリング確率を設定する手法と言える。双重み付けサンプリングでは信頼度に比例したサンプリング確率を設定し、クリーニングでは信頼度が閾値以下のデータのサンプリング確率を0、その他のサンプリング確率を定数に設定する。この関係は図 3.1 のように表せる。これらの大きな相違点は閾値前後

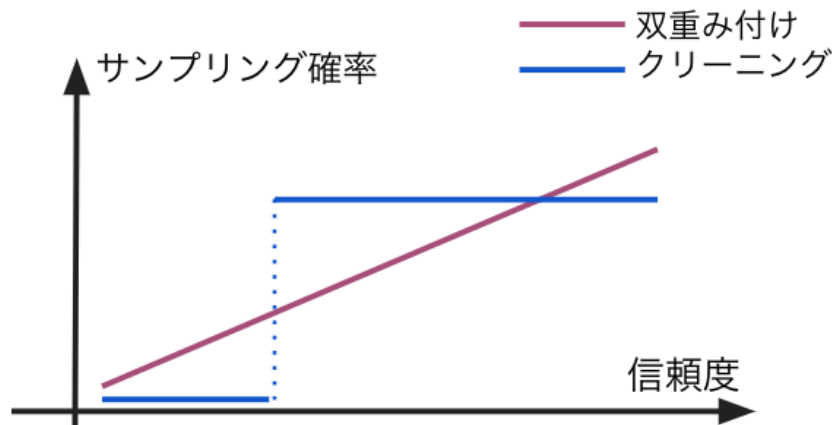


図 3.1. 2つの提案手法における信頼度に対するサンプリング確率の設定の概観。

のデータに関するサンプリング確率と言える。クリーニングにおいては閾値以下を全くサンプリングしない一方で、双重み付けサンプリングにおいてはその閾値以下のデータも少なからずサンプリングが行われる。また、クリーニングにおいては閾値以上は信頼度に依らず同じ確率

でサンプリングする一方で、多重み付けサンプリングにおいては閾値周辺のデータのサンプリング確率を低く設定する。これは 2 章でも述べたように、情報損失と極端にノイズなデータの影響とのトレードオフにある。ただし、CML への適用を考えるとときには、クリーニングのメリットが大きくなると予想される。それは、CML の損失関数におけるネガティブサンプルの扱いに起因している。式 (2.3) で表されるミニバッチに関する損失関数では、ポジティブサンプル B 内の各ユーザーアイテムペア (u, i) について抽出したネガティブサンプル $\mathcal{N}_{u,i}$ のうち、ユーザー u と最近傍に位置するネガティブアイテム $j \in \mathcal{N}_{u,i}$ に関する損失のみを考慮する。これは、図 3.2 で示される問題を引き起こす。この図 3.2 は、中心がユーザー u のベクトルの座標を表し、赤い丸がユーザー u とポジティブな関係を持つアイテム i の座標、そのほかの青い丸が $j_1 \sim j_5$ がユーザー u とポジティブな関係を持たないネガティブサンプル $\mathcal{N}_{u,i}$ のそれぞれの座標を表す。また、そのうち j_3 はネガティブノイズを表している。すなわち j_3 は潜在的にはユーザー u との嗜好が合うものの、ポジティブな関係は観測されていないとする。CML の性質上 j_3 はそれ自体が学習の対象にならずとも周辺のユーザーアイテムについての学習が進むにつれてユーザーと近接するように学習されるはずである。これによってユーザーと潜在的に嗜好の合うアイテムの推薦がうまく機能する。しかし、この j_3 が 1 度でもネガティブサンプルに含まれてしまうと、この図のように $j_3 = \operatorname{argmin}_{j \in \mathcal{N}_{u,i}} d(u, j)$ となりやすいため、 j_3 のみがユーザーから遠ざけるように学習されてしまう。これは、この j_3 が好ましくない方向に学習が進むだけでなく、全てのネガティブアイテム $\mathcal{N}_{u,i}$ について学習が進まない事を意味する。さらに、 j_3 の学習が好ましくない方向に学習されてしまうと、三角不等式の関係からその周辺のユーザーやアイテムについても悪影響を及ぼす。このように、CML にお

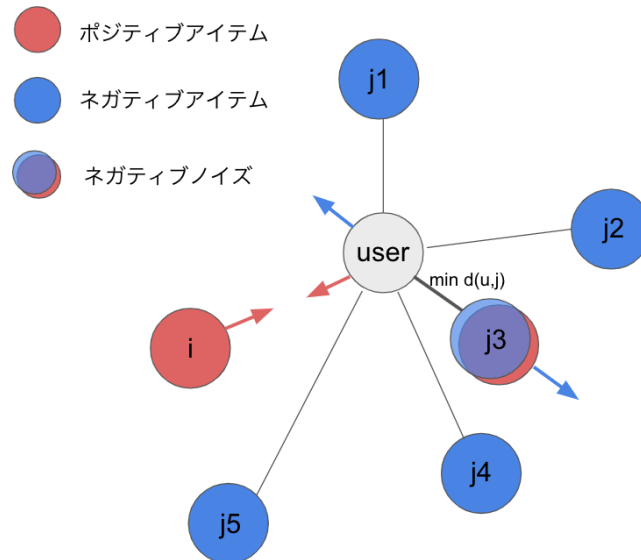


図 3.2. ネガティブサンプルにネガティブノイズが含まれる場合のミニバッチにおける学習の様子。

いてはノイズは少しでも含まれるとそのみが学習されてしまう。よって、信頼度が低いデー

タはサンプリング確率を小さくするだけでは不十分であり、適切な閾値を決めて学習の対象から削除するべきと考えられる。

以上より、2つの提案手法は情報損失およびノイズの影響におけるトレードオフを持つものの、CML への適用を考えるとクリーニングのメリットの方が大きくなる可能性が高いのではないかと予想される。これらの性能差についても続く数値実験において検証を行う。

3.3.3 大規模データへの応用可能性

次に、提案手法の大規模データに対する適用可能性について議論する。大規模データへの適用において重要なのは時間計算量と空間計算量である。一般的な推薦システムの文脈において大規模データへの適用を前提とする際、どちらの計算量も $\mathcal{O}(|S|)$ が許容されてきた。よって、提案手法もこれに則っているかを確認する必要がある。本研究の 2 つ提案手法は、大規模データへ適用可能なアルゴリズム CML について、その構成要素

- ミニバッチのサンプリング
- 損失関数の計算
- パラメータの更新

の中の 1 点目についてのみ改善したものである。したがって、その他の 2 点についての時間・空間計算量は標準的な CML と全く同じである。よって以下では、1 点目のミニバッチのサンプリングにおける計算量に注目する。

時間計算量について、提案手法が標準的な CML と異なる点は信頼度の推定によるサンプリング確率の計算を行う点と重み付きサンプリングへ変更する点である。信頼度の推定の計算量は信頼度の推定モデル $\tilde{f}(u, i | \tilde{\Theta})$ の設定によって異なるが、本研究の数値実験で採用したロジスティック行列分解の計算量は標準的な CML と変わらない。よって、提案手法の時間計算量は高々 2 倍程度の悪化であるため、ここでは許容できるものとする。また、一様サンプリングの時間計算量が抽出サンプル数 m において $\mathcal{O}(m)$ であるのに対し、重み付きサンプリングは全体候補数 n 、抽出サンプル数 m に対して $\mathcal{O}(m \log n)$ の時間計算量を要する。したがって双重み付けサンプリングでは、ポジティブサンプリングで $\mathcal{O}(|\mathcal{B}| \log |S|)$ 、ネガティブサンプリングでは $\mathcal{O}(|\mathcal{B}| \cdot |\mathcal{N}_{u,i}| \log |I|)$ の計算量が必要となる。ただし、 \mathcal{B} や $\mathcal{N}_{u,i}$ のサイズは十分に小さく設定するため、提案手法も時間計算量は許容できると言える。以上、各手法のサンプリングにおける時間計算量についてまとめた表が表 3.2 である。

表 3.2. 各手法における，ポジティブサンプリングおよびネガティブサンプリングの時間計算量オーダーを示す表

	ポジティブサンプリング	ネガティブサンプリング
標準的な CML	$\mathcal{O}(\mathcal{B})$	$\mathcal{O}(\mathcal{B})$
双重み付け	$\mathcal{O}(\mathcal{B} \log \mathcal{S})$	$\mathcal{O}(\mathcal{B} \cdot \mathcal{N}_{u,i} \log \mathcal{I})$
クリーニング	$\mathcal{O}(\mathcal{B})$	$\mathcal{O}(\mathcal{B})$

一方，空間計算量について標準的な CML と異なる点はサンプリング確率の記憶である．まずポジティブサンプリングについてであるが，重みの記憶が必要な手法は提案手法の双重み付けサンプリングのみである．したがって，ポジティブサンプリングに必要な空間計算量は双重み付けサンプリングのみが $\mathcal{O}(|\mathcal{S}|)$ であり，そのほかは $\mathcal{O}(1)$ である．ネガティブサンプリングについては，まずどの手法もユーザー u とインタラクションを観測していないアイテムの集合 \mathcal{D}_u を対象にサンプリングが行われる．そのため，単純に考えれば任意のユーザーについてインタラクションをしていないアイテムを記憶する必要があるが， $\sum_{u \in \mathcal{U}} |\mathcal{D}_u|$ だけ空間計算量が必要になる．しかし，一般に \mathcal{S} はスパースでありユーザー u とインタラクションをしたアイテム集合 $\mathcal{S}_u = \{i | Y_{u,i} = +1\} = \mathcal{I} \setminus \mathcal{D}_u$ はアイテム集合 \mathcal{I} と比較して十分小さい．よって，任意のユーザー u について \mathcal{D}_u を記憶するのではなく， \mathcal{S}_u を記憶する方が小さい容量で同様な目的を達成できる．すなわち，ネガティブサンプリングにおいて抽出してはいけない”NG リスト”として $\sum_{u \in \mathcal{U}} |\mathcal{S}_u| = |\mathcal{S}|$ だけ記憶すれば良い．それに加え，双重み付けサンプリングでは，ネガティブサンプリングを実行する前に，ポジティブサンプル \mathcal{B} 内の任意のユーザーアイテムペア $(u, i) \in \mathcal{B}$ について，どのアイテムをどれだけの確率でネガティブサンプルとして抽出するかといった確率を記憶する必要がある．よって，双重み付けサンプリングのネガティブサンプリングでは，”NG リスト”の $|\mathcal{S}|$ に加え $|\mathcal{B}| \cdot |\mathcal{I}|$ だけ空間計算量が必要である．また，クリーニングにおいてはノイズなデータを”NG リスト”に加える形で実装できる．ネガティブサンプリングにおいて削除するユーザーアイテムペア数は，削除割合 q に対して $q \cdot |\mathcal{D}_u| = q \cdot (|\mathcal{U}| \cdot |\mathcal{I}| - |\mathcal{S}|)$ となるため，”NG リスト”もこの分だけ拡大する．よって，クリーニングにおいてネガティブサンプリングに必要な空間計算量は $|\mathcal{S}| + q \cdot (|\mathcal{U}| \cdot |\mathcal{I}| - |\mathcal{S}|) = (1 - q) \cdot |\mathcal{S}| + q \cdot |\mathcal{U}| \cdot |\mathcal{I}|$ となる．以上，各手法のサンプリングにおける空間計算量についてまとめた表が表 3.3 である．この表の通り，空間計算量についても

表 3.3. 各手法における，ポジティブサンプリングおよびネガティブサンプリングの空間計算量オーダーを示す表

	ポジティブサンプリング	ネガティブサンプリング
標準的な CML	$\mathcal{O}(1)$	$\mathcal{O}(\mathcal{S})$
双重み付け	$\mathcal{O}(\mathcal{S})$	$\mathcal{O}(\mathcal{S} + \mathcal{B} \cdot \mathcal{I})$
クリーニング	$\mathcal{O}(1)$	$\mathcal{O}((1 - q) \cdot \mathcal{S} + q \cdot \mathcal{U} \cdot \mathcal{I})$

ポジティブサンプリングは全く問題がないと言える。一方、ネガティブサンプリングについては注意が必要である。多重み付けサンプリングでは、アイテム I が大きい場合ポジティブサンプル B のサイズを小さくしなくてはならない。確率的勾配降下法においてはミニバッチのサイズが大きいほど学習が安定的になることが多いためこの点については標準的な CML と比較した1つのデメリットであると言える。また、クリーニングについては、削除割合 q の影響を受ける。特にユーザー数およびアイテム数が膨大であるとき計算量 $\mathcal{O}(|U| \cdot |I|)$ は現実的に実行不可能となる。よってスパース性 $r = \frac{|S|}{|U| \cdot |I|}$ と同程度の削除割合 q を設定する必要がある。仮に $q = r$ とした場合、空間計算量は $\mathcal{O}((2 - r) \cdot |S|) = \mathcal{O}(|S|)$ となるため、大規模データでも許容しやすいと考えられる。

以上より、提案手法は大規模データへの適用も可能と考えられるが、ミニバッチのサイズやデータ削除割合の設定には注意が必要であると言える。

第 4 章

数値実験

本章では 3 章で示した提案手法の性能を実験的に検証する．特に，1 章で示したリサーチクエッションにしたがって，既存手法と比較した提案手法の性能改善を計測する．また，本実験は 2021 年 4 月から 5 月に Colaboratory の GPU 通常メモリインスタンス上で行ったが，今回使用した実験環境で GPU は常に NVIDIA Tesla V100 であったことを確認している．

4.1 データセット

本実験では，内在するバイアス/ノイズの影響を考慮して推薦システムを評価できることで知られる Yahoo! R3 データセットと coat データセットを用いた．これらのデータセットは，訓練データとテストデータに分かれており，訓練データは各ユーザーが自由に選択したアイテムの評価，テストデータは各ユーザーに対して完全にランダムに選択されたアイテムの評価を持っている．つまり，訓練データは一般的な推薦システム用のデータセットと同様にノイズを含んでいる一方で，テストデータでは，ノイズがほとんど発生していないと言える．これらのデータセットに共通の前処理を行うことで implicit feedback のデータセットを作成する．そして，訓練データで学習し，テストデータで評価することで，ノイズを含む implicit feedback のデータにおいてもユーザーの嗜好性を正確に学習できるかを検証できる．

前処理は以下の通りである．

- まず，これらのデータは 5 段階評価のデータであるため，訓練データおよびテストデータにおいて以下の処理を行うことで，バイナリの評価値として変換した．
 - 評価値が 4 以上の (u, i) について $R_{u,i} = +1$,
 - 評価値が 3 以下の (u, i) について $R_{u,i} = -1$,
 - 評価値が観測されていない (u, i) の $R_{u,i}$ は欠損とする．
 ただしここで，訓練データに含まれる評価を $R_{u,i}^{\text{train}}$ ，テストデータに含まれる評価を $R_{u,i}^{\text{test}}$ と表す，
- 次に，以下の処理によって，訓練データにおいて観測されるラベル $Y_{u,i}$ を作成する．

- 評価値が観測された全てのユーザーアイテムペア (u, i) について $Y_{u,i} = +1$
- その他を $Y_{u,i} = -1$

これら 1,2 の処理によって、訓練データに 2 種類のノイズが含まれると言える。

- ポジティブノイズ: $Y_{u,i} = +1$ かつ $R_{u,i} = -1$ (評価値が 3 以下)
- ネガティブノイズ: (訓練データ内では $Y_{u,i} = -1$ で、) テストデータ内で $R_{u,i} = +1$

一方で、検証時にはテストデータの真の評価値 $R_{u,i}^{\text{test}}$ を用いることで、ノイズを含まずに評価が可能となる。以上の前処理によって、得られたデータセットの簡単な特徴は表 4.1 で表される。

表 4.1. 前処理を施した後のデータセットの特徴

データセット	ユーザー数	アイテム数	インタラクション数
Yahoo! R3	15,400	1,000	311,704
coat	290	300	6960

4.2 比較手法

次に、本実験で比較する手法について整理する。ただし、すべての手法は CML をベースとし、サンプリングのみについて変更するものとする。まず、本実験では以下の既存手法をベースラインとする。

- uniform : 両サンプリングにおいて完全に一様なサンプリング
- 1stage : ネガティブサンプリングにおいて 1 段階の重み付け (2.2.5 節の通り)
- 2stage : ネガティブサンプリングにおいて 2 段階の重み付け (2.2.6 節の通り)

そして、1 つ目の提案手法である二重み付けサンプリングおよびそのアブレーションスタディとして以下を検証の対象とする。

- pos weight: ポジティブサンプリングのみ信頼度で重み付け
- neg weight: ネガティブサンプリングのみ信頼度で重み付け
- both weight: 二重み付けサンプリング (1 つ目の提案手法)

さらに、1 つ目の提案手法であるクリーニングおよびそのアブレーションスタディとして以下を検証の対象とする。

- pos clean : ポジティブサンプリングのみクリーニング
- neg clean : ネガティブサンプリングのみクリーニング
- both clean : 両サンプリングにおいてクリーニング (2 つ目の提案手法)

以下より，本章では各手法を上記の表記にて扱う．

4.3 ハイパーパラメータ

本実験では，全ての学習において，確率的勾配降下法の学習率を $1e-3$ ，埋め込みベクトル \mathbf{v}_* の次元 l を 10 とした．また，Yahoo! R3 データセットの実験では，信頼度推定モデルとして採用したロジスティック行列分解において 2 万回，CML において 5 万回のイテレーションを行った．一方，coat データセットの実験では，信頼度推定モデルのロジスティック行列分解において 3 千回，CML において 1 万回イテレーションを行った．さらに，本実験を通して， $|\mathcal{B}| = 256$ とし，Yahoo! R3 データセットでは $|\mathcal{N}_{u,i}| = 10$ ，coat データセットでは $|\mathcal{N}_{u,i}| = 5$ とした．そして最後に，2 つ目の提案手法であるクリーニングにおいて，データ削除する割合は $p = q = 5\%$ とした．

4.4 結果

以上の設定によって実行した各手法の性能を示し，その結果を考察する．1 章で示した 2 つのリサーチクエッションにしたがって，まずは (ユーザー–アイテム) 間の関係性の正確度として推薦性能について評価し，次に (ユーザー–ユーザー) 間および (アイテム–アイテム) 間の関係性の正確度として埋め込みベクトルの説明性について評価する．

4.4.1 推薦性能の評価

推薦の評価指標 まず，本実験で使用する推薦性能の評価指標について整理する．本実験では，CML によって得られる (ユーザー–アイテム) 間の距離に基づいた推薦がどれだけ機能するかを評価する．すなわち，各ユーザー u について，ユークリッド距離に基づいて test データ内のアイテムを並べ替えたときに， $R_{u,i} = +1$ であるアイテムがどれだけ上位にあるかを表すランキング指標を用いて評価する．ランキング指標は同データセットによる既存研究に習い，以下の 3 つを採用した．

$$\begin{aligned} \text{DCG@K} &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u^{\text{test}}|} \sum_{i \in \mathcal{S}_u^{\text{test}}} \frac{\mathbb{I}\{\hat{Z}_{u,i} \leq K\}}{\log(\hat{Z}_{u,i} + 1)} \\ \text{MAP@K} &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u^{\text{test}}|} \sum_{i \in \mathcal{S}_u^{\text{test}}} \sum_{k=1}^K \frac{\mathbb{I}\{\hat{Z}_{u,i} \leq k\}}{k} \\ \text{Recall@K} &= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u^{\text{test}}|} \sum_{i \in \mathcal{S}_u^{\text{test}}} \mathbb{I}\{\hat{Z}_{u,i} \leq K\} \end{aligned}$$

ただし，ここで $\hat{Z}_{u,i}$ はユーザー u にとってのアイテム i の予測順位を表し， $\mathcal{S}_u^{\text{test}} = \{i | R_{u,i}^{\text{test}} = +1\}$ を表す．また，DCG@K は理想的な予測順位の時の値で正規化した nDCG@K を用い

る．さらに，本研究では $|\mathcal{S}_u^{\text{test}}| = 0$ となるユーザーは，そもそも推薦によって体験を向上させることはできないと判断して評価対象から削除した．同データセットによる既存研究はこのようなユーザーに対するスコアを0としていたため，本実験のスコアの方が全体的に高い値が算出されているが，大小関係は損なっていない．

推薦性能の評価結果と考察 続いて，4.2節で示した各手法について計算したランキング指標および学習にかかった時間を示す．表4.2はYahoo! R3 データセットにおける全手法の評価指標のスコアおよび実行時間（ノイズ率推定を含む）を表している．同様に表4.3はcoatデータセットにおけるスコアおよび実行時間である．ただし，これらのスコアは訓練データをランダムに1割削除して一連の学習を行い，各評価指標を計算するという操作を5回行ったときの平均値を表している．また，これらの表において太字は最高スコアを表し，下線は2番目のスコアを表す．さらに，図4.1および図4.2は，それぞれ学習経過における手法ごとのnDCG@3, Recall@5のスコアの平均値および信頼区間を表している．

表4.2. Yahoo! R3 データセットにおける各手法のランキング指標と実行時間の結果．

	nDCG@3	MAP@3	Recall@3	nDCG@5	MAP@5	Recall@5	実行時間 [s]
uniform	0.494	0.550	0.523	0.567	0.564	0.702	3.90×10^2
1stage	0.450	0.493	0.502	0.534	0.516	0.704	4.26×10^2
2stage	0.416	0.459	0.464	0.505	0.487	0.680	19.05×10^2
pos weight	0.539	0.591	0.572	0.609	0.600	0.746	6.02×10^2
neg weight	0.530	0.585	0.559	0.599	0.594	0.730	6.51×10^2
both weight	0.541	0.596	0.571	0.613	0.605	0.748	6.60×10^2
pos clean	0.551	0.604	0.582	0.620	0.611	0.755	7.21×10^2
neg clean	0.559	0.612	0.590	0.628	0.620	0.763	7.43×10^2
both clean	0.559	0.612	0.590	<u>0.626</u>	<u>0.619</u>	<u>0.760</u>	7.53×10^2

表4.3. coat データセットにおける各手法のランキング指標と実行時間の結果

	nDCG@3	MAP@3	Recall@3	nDCG@5	MAP@5	Recall@5	実行時間 [s]
uniform	0.271	0.355	0.219	0.312	0.371	0.337	0.61×10^2
1stage	0.253	0.326	0.209	0.296	0.349	0.333	0.63×10^2
2stage	0.296	0.390	0.234	0.347	0.416	0.375	0.62×10^2
pos weight	0.341	0.446	0.282	0.393	0.466	0.432	1.00×10^2
neg weight	0.347	0.453	0.294	0.394	<u>0.469</u>	<u>0.439</u>	1.02×10^2
both weight	0.347	0.445	0.289	0.394	0.467	0.431	1.02×10^2
pos clean	0.340	0.448	0.281	0.393	0.462	0.432	1.19×10^2
neg clean	<u>0.346</u>	0.449	<u>0.290</u>	<u>0.397</u>	0.467	0.438	1.06×10^2
both clean	<u>0.346</u>	<u>0.450</u>	<u>0.290</u>	0.402	0.471	0.450	1.18×10^2

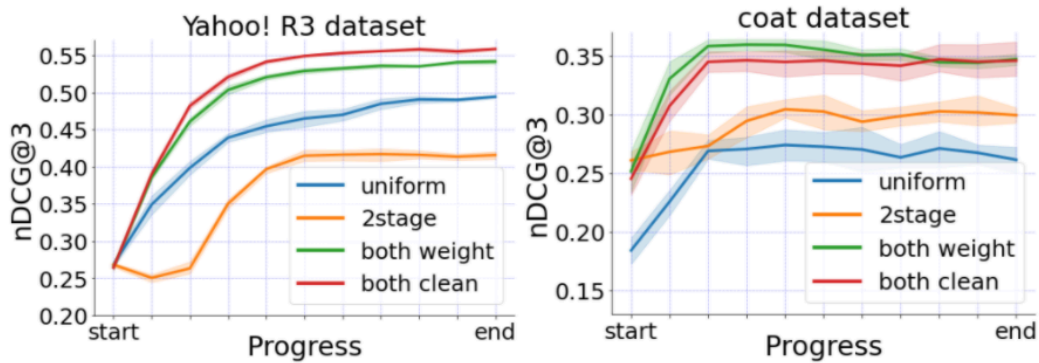


図 4.1. 各手法の学習経過における nDCG@3 の平均値と信頼区間. 横軸は CML の学習を始めてから終わりまでの進捗割合を示す. (左): Yahoo! R3 データセットにおける結果. (右): coat データセットにおける結果.

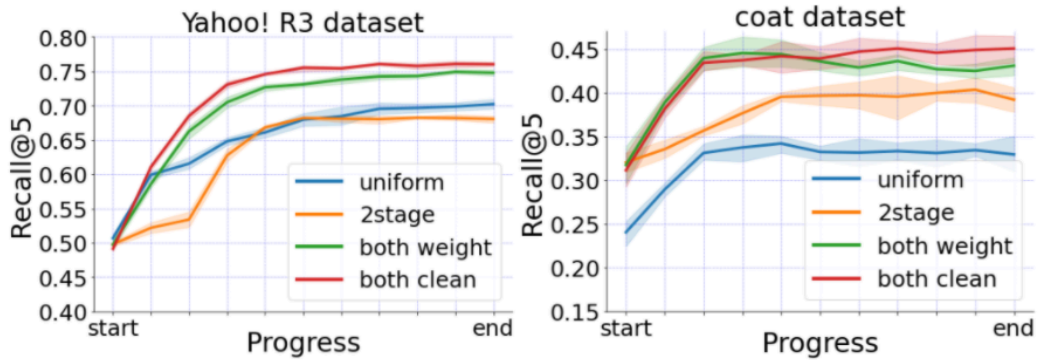


図 4.2. 各手法の学習経過における Recall@5 の平均値と信頼区間. 横軸は CML の学習を始めてから終わりまでの進捗割合を示す. (左): Yahoo! R3 データセットにおける結果. (右): coat データセットにおける結果.

表 4.2 および表 4.3 の結果を見て分かる通り, 信頼度やノイズ率を考慮した 6 つの手法は, 全ての指標において *uniform* などのベースラインから大幅に精度が向上している. また, 図 4.1, 図 4.2 において, どの手法も学習終了時の信頼区間は小さく, 提案手法による精度改善は有意な差と言える. さらに, これらの結果より, 全ての手法は設定したイテレーション数で収束しており, 学習時間を伸ばしてもこれ以上の改善は見込まれないと考えられる. よって, 表 4.2 および表 4.3 の実行時間の結果より, 提案手法の精度改善は *uniform* のおよそ 2 倍の計算時間を許容することで達成できると言える.

より詳しく見ていくと, Yahoo! R3 データセットでは *neg clean* および *both clean* が最も性能が良い, もしくは 2 番目に性能が良い手法であると言える. また, coat データセットでは, *neg weight* および *both clean* が最も性能が良い, もしくは 2 番目に性能が良い手法であると言える. 一方, Yahoo! R3 データセットにおける最高性能の *neg clean* は coat データセットにおいては良い性能が示されていない, coat データセットにおける最高性能の *neg*

weight は Yahoo! R3 データセットにおいては良い性能が示されていない。したがって、安定的に精度が高いのはポジティブノイズもネガティブノイズについてもデータを削除した *both clean* であると考えられる。これは、ポジティブノイズおよびネガティブノイズの両者において、重み付けをするのではなく、極めてノイズとみなせるデータを除去することがより有効であることを示している。

ここで、Yahoo! R3 データセットにおいて、*1stage* や *2stage* が *uniform* よりもスコアが悪化しているのはポジティブノイズの影響だと考えられる。これらの手法の有効性を確かめた既存研究はネガティブノイズにのみフォーカスしたため、 $Y_{u,i} = +1$ のラベルはそのユーザーとアイテムが確実に $R_{u,i} = +1$ であることを仮定していた。そして、そのラベルを用いて人気度が計算されるため、正確な人気度を入手可能だった。しかし、本実験ではポジティブノイズが含まれているため、 $R_{u,i} = +1$ とは異なる $Y_{u,i} = +1$ を用いて人気度を計算しており、本実験では正確な人気度を計算することができなかったと考えられる。これは実世界においても、implicit feedback を信頼して人気度を計算することが有効ではないことを示唆している。

4.4.2 埋め込みベクトルの説明性の評価

次に、(ユーザー–ユーザー) 間および (アイテム–アイテム) 間の関係性の正確さを評価する。特に、埋め込みベクトルがユーザーやアイテムの特徴を表現するかという説明性に注目する。そのために、同様な属性を持つユーザーおよびアイテムが距離の近い位置に埋め込みできているのか評価する。この実験では属性として *coat* データセットにおけるユーザーの性別および、アイテムの対象性別の情報を用いる。ただし、このデータセットにおけるアイテムは洋服を表す。図 4.3 は、*uniform* および *both clean* が学習して得られたユーザーの埋め込みを t-SNE [39] で 2 次元に圧縮し、それぞれの性別で色分けした散布図である。また、図 4.4 も同様にアイテムの埋め込みを対象性別で色分けした散布図である。洋服においては性別を超えたインタラクションは少ないため、属性を表現しやすいと考えられるものの、この図から *uniform* が生成した埋め込みは理想的でない部分があることが見て取れる。

より定量的に評価するため、SVM [40] を用いてどれだけ属性を分類できるかを評価した。すなわち、ユーザーおよびアイテムの埋め込みを特徴量に、属性のラベルを分類したときのエラー率を計算する。SVM はベクトル空間におけるサンプル間の距離をもとにラベルを予測する分類器であるため、CML で得られる埋め込みを評価するのに適したアルゴリズムである。表 4.4 は、5 分割交差検証によって分類のエラー率を評価した結果を表す。左の 2 列がユーザーの分類、右の 2 列がアイテムの分類についてであり、それぞれの左がエラー率の平均、右がエラー率の最大値 (最悪ケース) の結果である。

この結果より、*both clean* のエラー率は *uniform* からユーザーに関しては $3/4$ 、アイテムに関しては $1/2$ 程度と大幅に改善していることがわかる。また、最悪ケースではより大きな改善が見られる。よって、提案手法 *both clean* は (ユーザー–ユーザー) 間および (アイテム–アイテム) 間の関係性をよりよく捉えていると考えられる。この結果は、提案手法が現代の推薦シ

システムの要求に応えることができていることを意味する。

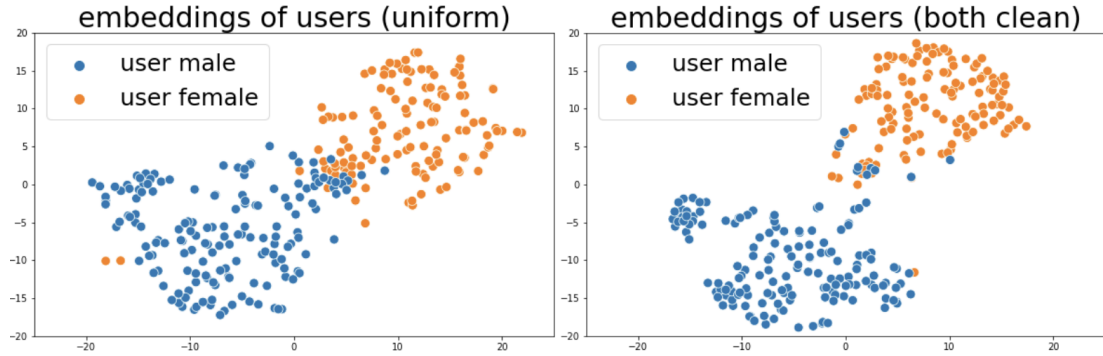


図 4.3. ユーザーの埋め込みを性別ごとに色分けして可視化した散布図. (左): *uniform* を用いて得られたユーザーの埋め込み. (右): *both clean* を用いて得られたユーザーの埋め込み.

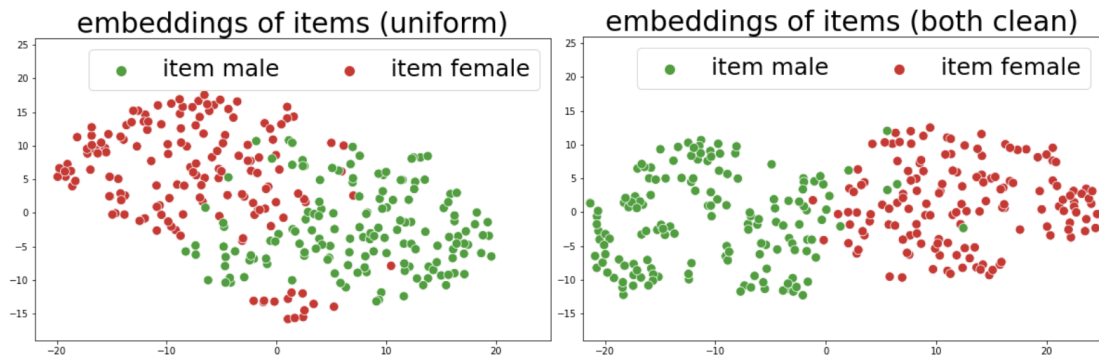


図 4.4. アイテムの埋め込みを対象性別ごとに色分けして可視化した散布図. (左): *uniform* を用いて得られたアイテムの埋め込み. (右): *both clean* を用いて得られたアイテムの埋め込み.

表 4.4. 埋め込みを特徴量として属性のラベルを SVM で分類したときの 5 分割交差検証のエラー率 [%] の結果

	User		Item	
	Avg.	Max.	Avg.	Max.
uniform	7.24	12.07	1.67	5.00
both clean	5.52	8.62	0.67	1.67

第5章

結論

最後に、本研究をまとめ、今後の課題を述べる。

本研究では、1) 嗜好性や類似性の学習時に implicit feedback の利用する、2) 嗜好性や類似性の予測・推論時に埋め込みベクトルを利用する、という web サービスの推薦システムにおける2つの要求に応えるアルゴリズムを提案した。具体的には、後者の要求を満たすために開発された協調距離計量学習 (CML) において、前者の要求である implicit feedback に起因するノイズを包括的に考慮したサンプリング方法を導入した。ノイズへ考慮する方法として提案した手法は2通りある。1つ目の手法はポジティブサンプリングおよびネガティブサンプリングの両者において事前に推定したデータの信頼度で重み付けする「二重重み付けサンプリング」であり、2つ目の手法は、事前に推定したノイズ率が高いデータをサンプリングの対象から削除する「クリーニング」である。これらの2つの手法は、理論的には後者の方が優れている性質があるものの、一般的な情報損失とノイズへの脆弱性によるトレードオフを内包するため、両者を現実世界で収集したデータセットにおいて検証した。その結果、2つの手法はどちらも1) および2) の2つの要求において既存研究を大幅に上回る性能を示した。また、本研究の数値実験において使用したデータセットでは予想通り2つ目の提案手法が安定的な性能を発揮することがわかった。

今後の課題は主に4つある。

1つ目の課題は、不偏性や一致性など統計的に優れた性質を持った $P(R_{u,i} = \pm 1)$ の推定量の開発である。本研究では、 $P(R_{u,i} = \pm 1)$ の代替として $P(Y_{u,i} = \pm 1)$ を推定したが、3.3節で示すように潜在的なノイズ率が高い場合バイアスが発生する。また、一致性についても示すことができていない。よって、 $Y_{u,i}$ の観測値のみから計算可能な優れた推定量が開発できればより適切なノイズ処理が可能と考えられる。

2つ目の課題は、サンプリング確率のさらなる工夫である。本研究では、推定した信頼度按比例した重み付け、もしくは閾値以下の信頼度を持つデータを除外しその他を一様に抽出する方法をとった。しかし、信頼度に対数をかけて重み付けを行う設定や、閾値以下は除外し閾値以上は線形的な重み付けを行う設定など、サンプリング確率の設定はまだ工夫の余地があると言える。よって、理論的・実験的にノイズに頑健なサンプリング確率の設定方法の確立は今後

の課題と言える。

3 つ目の課題は提案手法クリーニングにおける最適なデータの削除割合 p, q の決定である。本研究ではこれらをひとつハイパーパラメータとして任意に決定したが、より実用的には情報損失とノイズへの脆弱性を鑑みた最適な削除割合の決定が望まれる。

4 つ目の課題が実サービスへの実装である。本研究では、現実世界で収集したノイズを含まないデータセットにおいてその性能を検証した。しかしながら、ノイズを含まないデータセットの収集コストは高く、現代 web サービスの規模と比較して極めて小さいものであった。また、これらのデータを用いた評価指標の改善がどれだけの経済効果を創出するかは不明である。よって、本研究の提案手法の本質的な価値を検証するには、実サービスにおいて実装し収益性やアクティブユーザー数などの実際のサービスに関わる KPI への寄与度を計測する必要があると言える。

謝辞

本論文を完成させるにあたって、多くの方々にご協力いただきました。なによりもまずこの素晴らしい研究環境を用意していただいた担当教員の中田和秀教授に心から感謝申し上げます。また、本論文の課題は SMN 株式会社 a.i.lab にインターン生として在籍していた頃に業務として取り組ませていただいた内容から発展した形となりました。メンターとしてご指導頂き、国際学会の論文では共著者に入っていたいただいた柳沼傑さんをはじめ、みなさまから厚いご支援をいただきました。誠に感謝いたします。また、中田研究室の OB・OG や先輩方、同輩、後輩の質の高い研究を目の当たりにすることで、気を引き締めて研究にとりかかることができました。大変充実した研究生生活をともにできてとても嬉しく思います。

最後に、さまざまな苦勞をかけ、この私を大学院卒業まで育て、見守ってくれた両親および祖父母・兄への多大なる感謝を記し、本論文を締めさせていただきます。

2022 年 2 月 松井 諒生

参考文献

- [1] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blak Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pp. 293–296, 2010.
- [2] Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 311–320, 2018.
- [3] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at spotify. In *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*, p. 373, 2016.
- [4] Xavier Amatriain and Justin Basilico. Recommender systems in industry: A netflix case study. In *Recommender Systems Handbook*, pp. 385–419. Springer, 2015.
- [5] Chao-Yuan Wu, Christopher V Alvino, Alexander J Smola, and Justin Basilico. Using navigation to improve recommendations in real-time. In *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 341–348, 2016.
- [6] Fernando Amat, Ashok Chandrashekar, Tony Jebara, and Justin Basilico. Artwork personalization at netflix. In *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 487–488, 2018.
- [7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW '17 : Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.
- [8] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI '09: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, 2009.
- [9] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 191–198, 2016.

- [10] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–344, 2017.
- [11] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 501–509, 2020.
- [12] Yuta Saito. Unbiased pairwise learning from biased implicit feedback. In *ICTIR'20 : Proceedings of the 2020 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 5–12, 2020.
- [13] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR '21: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [14] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *WSDM '21: Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 373–381, 2021.
- [15] Wenhui Yu and Zheng Qin. Sampler design for implicit feedback data by noisy-label robust learning. In *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 861–870, 2020.
- [16] Nan Wang, Zhen Qin, Xuanhui Wang, and Hongning Wang. Non-clicks mean irrelevant? propensity ratio scoring as a correction. In *WSDM '21: Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 481–489, 2021.
- [17] Patrick Wieschollek, Oliver Wang, Alexander Sorkine-Hornung, and Hendrik Lensch. Efficient large-scale approximate nearest neighbor search on the gpu. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2027–2035, 2016.
- [18] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based retrieval in facebook search. In *KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561, 2020.
- [19] Dongyan Guo, Jingsong Xu, Jian Zhang, Min Xu, Ying Cui, and Xiangjian He. User

- relationship strength modeling for friend recommendation on instagram. *Neurocomputing*, Vol. 239, pp. 9–18, 2017.
- [20] Yuri M. Brovman, Marie Jacob, Natraj Srinivasan, Stephen Neola, Daniel Galron, Ryan Snyder, and Paul Wang. Optimizing similar item recommendations in a semi-structured marketplace to maximize conversion. In *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 199–202, 2016.
- [21] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *WWW '17 : Proceedings of the 26th International Conference on World Wide Web*, pp. 193–201, 2017.
- [22] Yifan Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008.
- [23] Lu Yu, Chuxu Zhang, Shichao Pei, Guolei Sun, and Xiangliang Zhang. Walkranker: A unified pairwise ranking model with multiple relations for item recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pp. 2596–2603, 2018.
- [24] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang. Collaborative similarity embedding for recommender systems. In *WWW '19: The World Wide Web Conference*, pp. 2637–2643, 2019.
- [25] Miguel Campo, JJ Espinoza, Julie Rieger, and Abhinav Taliyan. Collaborative metric learning recommendation system: Application to theatrical movie releases. *arXiv preprint arXiv:1803.00202*, 2018.
- [26] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Latent relational metric learning via memory-based attention for collaborative ranking. In *WWW '18: Proceedings of the 2018 World Wide Web Conference*, pp. 729–739, 2018.
- [27] Xiao Zhou, Danyang Liu, Jianxun Lian, and Xing Xie. Collaborative metric learning with memory network for multi-relational recommender systems. In *IJCAI'19: Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4454–4460, 2019.
- [28] Chanyoung Park, Donghyun Kim, Xing Xie, and Hwanjo Yu. Collaborative translational metric learning. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 367–376, 2018.
- [29] Lucas Vinh Tran, Yi Tay, Shuai Zhang, Gao Cong, and Xiaoli Li. Hyperml: A boosting metric learning approach in hyperbolic space for recommender systems. In *WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 609–617, 2020.
- [30] Viet-Anh Tran, Guillaume Salha-Galvan, Romain Hennequin, and Manuel Moussal-

- lam. Hierarchical latent relation modeling for collaborative metric learning. *arXiv preprint arXiv:2108.04655*, 2021.
- [31] Viet-Anh Tran, Romain Hennequin, Jimena Royo-Letelier, and Manuel Moussallam. Improving collaborative metric learning with efficient negative sampling. In *SIGIR'19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1201–1204, 2019.
 - [32] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, Vol. 70, pp. 1373–1411, 2021.
 - [33] Ryo Matsui, Suguru Yaginuma, Taketo Naito, and Kazuhide Nakata. Confident Collaborative Metric Learning. In *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 246–253, 2021.
 - [34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 26, pp. 3111–3119, 2013.
 - [35] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2016.
 - [36] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015.
 - [37] Christopher C Johnson. Logistic matrix factorization for implicit feedback data. In *Advances in neural information processing systems 27 : 28th Annual Conference on Neural Information Processing Systems 2014 (NIPS)*, pp. 1–9, 2014.
 - [38] Steffen Rendle. Factorization machines. In *ICDM '10: Proceedings of the 2010 IEEE International Conference on Data Mining*, pp. 995–1000, 2010.
 - [39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, Vol. 9, No. 11, 2008.
 - [40] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, Vol. 10, No. 5, pp. 988–999, 1999.