



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0011415
(43) 공개일자 2025년01월21일

(51) 국제특허분류(Int. Cl.)

G06N 20/00 (2019.01)

(52) CPC특허분류

G06N 20/00 (2021.08)

(21) 출원번호 10-2023-0091806

(22) 출원일자 2023년07월14일

심사청구일자 2023년07월14일

(71) 출원인

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

김태한

경기도 부천시 계남로 106, 406동 2401호

정원주

서울특별시 강남구 삼성로64길 5, 107동 302호

(74) 대리인

김동용

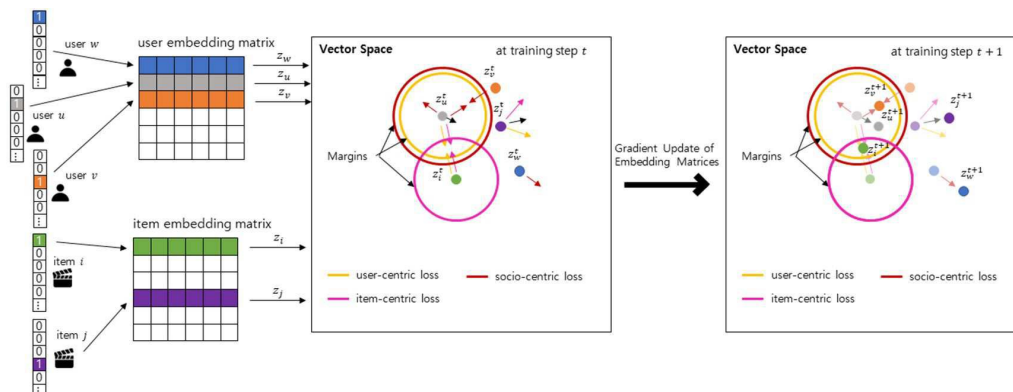
전체 청구항 수 : 총 4 항

(54) 발명의 명칭 신뢰 네트워크 상에서의 사회적 협업 메트릭 학습에 기반한 추천 장치 및 방법

(57) 요약

신뢰 네트워크 상에서의 사회적 협업 메트릭 학습 방법이 개시된다. 상기 메트릭 학습 방법은 적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되고, 유저들 각각과 아이템들 각각에 대한 임베딩 벡터(embedding vector)를 생성하는 단계, 손실을 계산하는 단계, 및 임베딩 벡터에 대한 그라디언트를 업데이트하는 단계를 포함하고, 상기 손실을 계산하는 단계와 상기 업데이트하는 단계는 복수회 반복 수행되고, 상기 손실을 계산하는 단계는 수학식 1에 의해 정의되는 손실 함수(L_{total})를 이용하여 계산된다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1711193973
과제번호	2019-0-00079-005
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성
연구과제명	인공지능대학원지원(고려대학교)
기 여 율	1/2
과제수행기관명	고려대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1345360494
과제번호	5199991014614
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	4단계두뇌한국21사업(R&D)
연구과제명	인공지능교육연구단
기 여 율	1/2
과제수행기관명	고려대학교
연구기간	2022.03.01 ~ 2023.02.28

명세서

청구범위

청구항 1

적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되는 메트릭 학습 방법에 있어서,
 유저들 각각과 아이템들 각각에 대한 임베딩 벡터(embedding vector)를 생성하는 단계;
 손실을 계산하는 단계; 및
 임베딩 벡터에 대한 그래디언트를 업데이트하는 단계를 포함하고,
 상기 손실을 계산하는 단계와 상기 업데이트하는 단계는 복수회 반복 수행되고,
 상기 손실을 계산하는 단계는 수학적 식 1에 의해 정의되는 손실 함수(L_{total})를 이용하여 계산되고,

$$\begin{aligned} \mathcal{L}_{total} = & \sum_{(u,i,j) \in \mathcal{D}} \max(m + d(u,i) - d(u,j), 0) \\ & + \lambda \sum_{(u,i,j) \in \mathcal{D}} \max(n + d(i,u) - d(i,j), 0) \\ & + \rho \sum_{(u,v,w) \in \mathcal{S}} \max(h + d(u,v) - d(u,w), 0) \\ & + \gamma \mathcal{L}_{AM} \end{aligned}$$

상기 수학적 식 1은 이고,

상기 $\mathcal{D}$ 와 상기 $\mathcal{S}$ 는 데이터 집합, 상기 u 는 유저, 상기 i 는 상기 u 의 긍정 아이템(positive item), 상기 j 는 상기 u 의 부정 아이템(negative item), 상기 $d(\cdot)$ 는 거리(distance), 상기 v 는 상기 u 의 트러스티(trustee), 상기 w 는 상기 u 의 트러스티가 아닌 유저, 상기 λ , ρ , 및 γ 는 가중치, 상기 m 은 제1 마진(margin), 상기 n 은 제2 마진, 상기 h 는 제3 마진, 그리고 상기 $\mathcal{L}_{AM}$ 은 적응형 마진 손실(adaptive margin loss)를 나타내는,

메트릭 학습 방법.

청구항 2

제1항에 있어서,
 적어도 하나의 유저와 아이템들 간의 거리가 계산되는 단계를 더 포함하는,
 메트릭 학습 방법.

청구항 3

제2항에 있어서,
 상기 거리는 유클리드 거리인,
 메트릭 학습 방법.

청구항 4

제3항에 있어서,
 가장 짧은 거리를 나타내는 K (K 는 임의의 자연수) 개의 아이템을 추출하는 단계를 더 포함하는,
 메트릭 학습 방법.

발명의 설명

기술분야

[0001] 본 발명은 추천 시스템에 관한 것으로, 보다 상세하게는 신뢰 네트워크 데이터, 예컨대 쇼핑 등의 아이템과 친구 맺기 등의 유저 간의 관계의 두 가지 관계를 모두 포함하는 데이터를 이용한 추천 시스템에 관한 것이다.

배경기술

[0002] 지난 수십 년 동안, 인터넷은 폭발적으로 확장되었다. 방대한 정보를 지닌 인터넷을 바탕으로 여러 분야의 문제를 해결하는 빅데이터 기술, 기계 학습, 그리고 딥 러닝 가짜 뉴스 탐지, 헬스 케어, 컴퓨터 비전, 그리고 추천과 같은 내용들이 소개되었지만 그에 따른 계산 부하는 여전히 부담으로 남아 있습니다. 한편, 인터넷 유저들은 많은 양의 다양한 정보 속에서 원하는 것을 선택하기 어려운 정보 과잉의 시대에 살고 있다. 따라서 유저가 원하는 내용 목록을 선제적으로 제공하는 것이 중요합니다.

[0003] 추천 시스템은 아마존, 유튜브, 넷플릭스와 같은 다양한 웹사이트에서 널리 사용되고 있다. Top-K 개인화 추천은 일반적으로 개별 유저의 요구를 만족시키기 위해 추천 시스템에 의해 수행된다.

[0004] 사회적 추천 시스템은 사람들 간의 관계인 사회적 네트워크라는 추가적인 정보를 사용하여 추천 성능을 향상시킨다. 신뢰 네트워크는 유저-아이템 네트워크와 유저-유저 관계를 모두 포함하는 사회적 네트워크이다. 신뢰 네트워크는 동종(Homophily)이라고 불리는, 유사한 취향을 가진 사람들이 자연스럽게 모이거나 연관시키는 특성을 가지고 있기 때문에 신뢰 네트워크에서 아이템 추천 성능의 향상은 다양한 논문과 실제 데이터로 검증되었다.

[0005] Top-K 추천에 대한 신뢰 네트워크의 기존 추천 방법은 별점 예측과 직접 랭킹의 두 가지 접근 방식을 기반으로 한다. 별점 예측 접근법은 유저의 별점이 없는 아이템에 대한 예측 별점을 제공하며, 예상 별점이 높은 Top-K 아이템을 개별 유저에게 추천한다. 그러나, 높은 별점의 Top-K 아이템이 최고의 Top-K 아이템 랭킹을 통한 추천과 항상 일치하는 것은 아니므로, Top-K 아이템 추천에는 직접 랭킹 방법이 선호된다. 또한, 별점 예측은 일반적으로 유저의 특정하고 명시적인 선호도를 나타내는 명시적인 데이터(예: 영화의 별점 1~5개)를 기반으로 하며, 클릭, 구매 및 보기와 같은 더 넓은 범위의 데이터를 포함하는 암시적 데이터에는 적용할 수 없다. 또한, 유저-유저 상호작용을 학습하는 과정에서, 여러 모델들은 계산 부담을 유발하는 유저-유저 간의 코사인 유사성 행렬을 활용한다.

[0006] 신뢰 네트워크의 기존 직접 랭킹 접근법은 유저와 아이템 간의 상관 관계(내적으로 계산)에서 특정 유저가 특정 아이템을 선호할 확률을 찾는 것을 기반으로 한다. 그러나, 이러한 내적 기반 방법은 근본적인 한계가 있다. 즉, 유사성 전파(similarity propogation) 과정의 부재로 인해 현재 기준이 되는 아이템과 유사한 아이템도 서로 유사한 점과 유저 간의 유사성을 제대로 학습하지 못한다. 또한, 이러한 모든 알고리즘은 네트워크의 아이템-아이템 관계를 무시하므로, 이를 활용한다면 랭킹 성능을 향상시킬 수 있다.

선행기술문헌

특허문헌

- [0007] (특허문헌 0001) 대한민국 공개특허 제2022-0075602호 (2022.06.08. 공개)
- (특허문헌 0002) 대한민국 공개특허 제2022-0147833호 (2022.11.04. 공개)
- (특허문헌 0003) 대한민국 공개특허 제2019-0056009호 (2019.05.24. 공개)
- (특허문헌 0004) 대한민국 공개특허 제2021-0146611호 (2021.12.06. 공개)

발명의 내용

해결하려는 과제

[0008] 별점 예측 후 이를 토대로 랭킹을 매기는 방식은 Top-K 아이템 추천의 최적의 방법이 아니다. 따라서, Top-K 아이템 추천을 위한 최적의 방법인 직접 랭킹 방법을 제안한다. 최근의 GNN(Graph Neural Network)이나 딥러닝(Deep Learning) 방법은 많은 계산을 필요로 하여 많은 컴퓨팅 리소스를 필요로 하기에 이를 사용하지 않은 간단한 알고리즘을 제안한다.

과제의 해결 수단

[0009] 본 발명의 일 실시예에 따른 메트릭 학습 방법은 적어도 프로세서(processor)를 포함하는 컴퓨팅 장치에 의해 수행되는 메트릭 학습 방법으로, 유저들 각각과 아이템들 각각에 대한 임베딩 벡터(embedding vector)를 생성하는 단계, 손실을 계산하는 단계, 및 임베딩 벡터에 대한 그래디언트를 업데이트하는 단계를 포함하고, 상기 손실을 계산하는 단계와 상기 업데이트하는 단계는 복수회 반복 수행되고, 상기 손실을 계산하는 단계는 수학적 식 1에 의해 정의되는 손실 함수(L_{total})를 이용하여 계산된다.

발명의 효과

[0010] MLP(Multi-Layer Perceptron)나 GNN 등의 구조 없이 3개의 삼중항 손실 (triplet loss)을 이용하여 직접 랭킹 방법을 구현할 수 있다. 복잡한 구조가 없기 때문에 컴퓨팅 리소스를 아낄 수 있고 보다 빠른 시간 안에 알고리즘 학습이 가능하다. 또한, 별점 예측을 수행한 다음 랭킹을 매기지 않고 직접 랭킹을 매기기에, Top-K 아이템 추천에 최적화된 알고리즘이라 할 수 있다.

도면의 간단한 설명

[0011] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.
도 1은 본 발명의 일 실시예에 따른 임베딩 벡터의 그래디언트 업데이트 과정을 설명하기 위한 개념도이다.
도 2는 본 발명의 일 실시예에 따른 메트릭 학습 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0013] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0014] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0015] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0016] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0017] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의

미로 해석되지 않는다.

- [0018] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.
- [0019] 본 발명에서는 CSML(Collaborative Social Metric Learning in Trust Network for Recommender Systems, 신뢰 네트워크 상에서의 사회적 협업 메트릭 학습)이라 명명될 수 있는 메트릭 학습을 기반으로 하는 신뢰 네트워크에서 Top-K 직접 랭킹 추천 시스템을 제안한다. 유저와 아이템은 d차원 벡터로 표현되고 삼중항 손실(triplet loss)을 사용하여 관련 있는 쌍은 밀집하게 모이게 되며, 관련 없는 쌍은 멀리 배치하게 된다. CSML은 추가적으로 유저와 유저 간의 관계인 유저-유저 관계를 고려하며, 관련이 있는 유저-유저 쌍은 신뢰 네트워크에 의해 결정된 "트러스터(trustor)-트러스티(trustee)" 관계를 공유하는 두 유저로 정의된다. 반면, 관련이 없는 유저-유저 쌍은 이 관계가 없는 유저 쌍을 의미한다. 결과적으로, CSML은 다음 세 가지 삼중항 손실의 합계(또는 가중합)를 사용하여 직접적인 Top-K 랭킹 아이템 추천을 달성할 수 있다. 첫째, 유저-아이템 관계를 학습하기 위한 유저 중심 손실(제1 손실), 두 번째, 아이템-아이템 관계를 학습하는 아이템 중심 손실(제2 손실), 마지막으로 유저-유저 관계를 학습하는 사회 중심 손실(제3 손실). 상술한 손실과 관련하여 메트릭 학습을 진행 후 유저와의 거리가 가장 가까운 K(K는 임의의 자연수)개의 아이템이 유저에게 추천될 수 있다.
- [0020] 우선, 추천 시스템에서 메트릭 학습 기반 알고리즘을 설명하기 위한 용어와 정의에 대해 설명한다.
- [0021] $\mathcal{U}$ 는 유저(user) 집합으로, $\mathcal{V}$ 는 아이템(item; 영화, 상품, 서비스 등을 의미할 수 있음) 집합으로 정의된다. $I := \{(u, v) | u \in \mathcal{U}, v \in \mathcal{V}\} \subseteq \mathcal{U} \times \mathcal{V}$ 는 상호 작용이 있는 모든 유저-아이템 쌍으로 정의되고, $I^- := \mathcal{U} \times \mathcal{V} \setminus I$ 는 다음과 같이 상호작용이 없는 유저-아이템 쌍으로 정의된다. 상호 작용이 있다는 것은 유저의 해당 아이템에 대한 평가(별점, 평점 등)가 존재하는 경우를 의미할 수 있다.
- [0022] 신뢰 네트워크(trust network)는 일반적으로 유저-유저 관계를 포함한 추가 정보로 특징지어 진다. 즉, 신뢰 네트워크는 유저와 아이템 간의 관계를 나타내는 유저-아이템 네트워크와 유저와 유저 간의 관계를 모두 포함하는 사회적 네트워크(social network)로 이해될 수 있다. 유저들 간의 관계인 유저-유저 관계(user-user relation)를 나타내는 행렬(matrix)을 T 라고 정의할 때, 사용자 i가 트러스터(trustor)이고 사용자 j가 트러스티(trustee)이면 T_{ij} 는 1의 값을 가지고, 그렇지 않다면(즉, 사용자 i와 사용자 j 간의 트러스터-트러스티 관계가 없다면) T_{ij} 는 0의 값을 갖는다. 이때, T 는 대칭이 아닐 수 있다. 즉, 트러스터-트러스티 관계(trustor-trustee relation)는 데이터에 따라 결정된다. 예를 들어, 어떤 유저가 기준이 될 때, 기준이 되는 유저는 트러스터이며 리스트에 등록했거나 팔로우 하는 유저들이 해당 유저의 트러스티가 될 수 있다.
- [0023] **메트릭 학습 기반 추천 시스템**
- [0024] 메트릭 학습 기반 추천 시스템(metric learning-based recommendation systems)에서, 모든 원소 $e \in \mathcal{U} \cup \mathcal{V}$ 는 유클리드 거리(Euclidean distance)를 제공한 d차원 메트릭 공간(metric space)에 임베딩된다(embedded)(Chanyoung, P., Donghyun, K., Xing, X., & Hwanjo, Y. (2018). Collaborative Translational Metric Learning. In Proceedings of the 2018 18th IEEE International Conference on Data Mining (ICDM '18). <https://doi.org/doi:10.1109/icdm.2018.00052>).
- [0025] z_a 와 z_b 가 각각 a and $b \in \mathcal{U} \cup \mathcal{V}$ 의 임베딩(embedding) 벡터를 나타낸다고 할 때, a와 b 사이의 거리(distance)는 다음과 같다.
- [0026]
$$d(a, b) = \|z_a - z_b\|^2$$
- [0027] 메트릭 학습 기반 추천 시스템의 목적은 특정 유저에 대해 관련된 아이템 i가 유저와 관련 없는 아이템보다 더 가깝게 위치하도록 유저 임베딩 벡터와 아이템 임베딩 벡터를 배치하고, 동시에 특정 아이템에 대해 관련 유저가 해당 항목과 관련 없는 아이템보다 더 가깝게 위치하도록 하는 것이다.
- [0028] 사용자 u가 주어졌을 때, 음이 아닌 마진(margin) m이 주어지고 모든 $(u, i) \in I$ 와 $(u, j) \in I^-$ 에 대해 다음이 성립해야 한다.

$$d(u, j) \geq d(u, i) + m$$

또한, 사용자 u 와 관련된 아이템 i 가 주어졌을 때, 음이 아닌 마진 n 이 주어지고 모든 $(u, i) \in I^+$, $(u, j) \in I^-$ 에 대해 다음이 성립해야 한다.

$$d(i, j) \geq d(i, u) + n$$

상술한 조건을 만족시키기 위해 다음과 같은 삼중항 손실(triplet losses)이 고려될 수 있다.

주어진 삼중항 (u, i, j) 에 대해, $(u, i) \in I^+$, $(u, j) \in I^-$ 이고, 음이 아닌 마진이 m 일 때 유저 중심 손실(user-centric loss)이라고 불리는 $\max(m+d(u, i)-d(u, j), 0)$ 를 채용하고, 이를 통해 임베딩 벡터의 그라디언트(gradient) 업데이트를 수행할 수 있다(Hsieh, C.-K., Yang, L., Cui, Y., Li, T.-Y., Belongie, S., & Estrin, D. (2017). Collaborative metric learning. In Proceedings of the 2017 World Wide Web Conference (WWW '17). ACM. <https://doi.org/doi:10.1145/3038912.3052639>). 삼각 부등식(triangle inequality)으로 인해 유저와 유사한 아이템들이 모이는 경향이 있으며, 결과적으로 유사성 전파 과정(similarity propagation process) 역시 성립하게 된다.

또한, 동일한 삼중항 (u, i, j) 에 대해 아이템 중심 손실(item-centric loss)이라고 불리는 $\max(n+d(i, u)-d(i, j), 0)$ 이 도입될 수 있으며(Li, M., Zhang, S., Zhu, F., Qian, W., Zhang, L., Han, J., & Hu, S. (2020). Symmetric Metric Learning with Adaptive Margin for Recommendation. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI '20). AAAI. <https://doi.org/doi:10.1609/aaai.v34i04.5894>), 유저 중심 손실과 같은 기능을 하게 된다.

유저 중심 손실의 각 사용자와 아이템 중심 손실의 각 항목에 대해 개인별로 적용되는 적응형 마진(adaptive margins)은 아이템 추천 시스템의 성능을 더욱 향상시킬 수 있다(Li et al., 2020). 개별 마진 m_u 와 n_i 를 최적화하기 위해 $0 \leq m_u, n_i \leq 1$ 의 제약 조건과 함께 다음 손실 함수가 추가적으로 사용될 수 있다.

$$\mathcal{L}_{AM} = - \left(\sum_{u \in \mathcal{U}} \frac{m_u}{|\mathcal{U}|} + \sum_{i \in \mathcal{I}} \frac{n_i}{|\mathcal{I}|} \right)$$

이하에서는, 본 발명의 실시예들에 대해 보다 구체적으로 설명한다.

신뢰 네트워크에서 트러스터와 트러스티는 유사한 취향이나 관심사를 가지고 있으므로, 트러스터의 임베딩 벡터와 트러스티의 임베딩 벡터는 메트릭 학습을 통해 학습된 메트릭 공간에서 서로 가까운 곳에 위치해야 한다. 즉, 유저를 지정할 경우 비연결(unconnected) 유저(트러스터도 트러스티도 아닌 유저) 보다 트러스티를 해당 유저에게 더 가깝게 배치하는 것이 바람직하다. 주어진 유저 u 에 대해 다음 부등식이 성립된다.

$$d(u, w) \geq d(u, v) + h$$

주어진 삼중항 (u, v, w) 에 대해 $u, v, w \in \mathcal{U}$ 이고, $T_{u, w} = T_{w, u} = 0$ 이며 $T_{u, v} = 1$ 이고 h 는 음이 아닌 마진이다.

이때, 사회 중심 손실(sociocentric-loss) $\mathcal{L}_s$ 는 $\max(h+d(u, v)-d(u, w), 0)$ 로 정의된다. 이 사회 중심 손실은 사용자 중심 손실 및 아이템 중심 손실과 동시에 최소화되어야 하기 때문에 최종 손실은 세 가지 손실과 $\mathcal{L}_{AM}$ 의 합계로 제공될 수 있다. 이때, 사회 중심 손실 역시 마진은 개인화 될 수 있다. 즉, 각 사용자 u 에 대해 개별 마진 $0 \leq h_u \leq 1$ 역시 고려될 수 있다. 이 경우 적응형 마진 손실 항 $\mathcal{L}_{AM}$ 은 다음과 같이 수정될 수 있다.

$$\mathcal{L}_{AM} = - \left(\sum_{u \in \mathcal{U}} \frac{m_u + h_u}{|\mathcal{U}|} + \sum_{i \in \mathcal{I}} \frac{n_i}{|\mathcal{I}|} \right)$$

여기서, $0 \leq m_u, n_i, h_u \leq 1$ 를 만족한다.

[0043] 마지막으로, 본 발명(CSML)에 대한 최종 손실 함수(loss function)는 다음과 같다.

$$\begin{aligned}\mathcal{L}_{total} = & \sum_{(u,i,j) \in \mathcal{D}} \max(m + d(u,i) - d(u,j), 0) \\ & + \lambda \sum_{(u,i,j) \in \mathcal{D}} \max(n + d(i,u) - d(i,j), 0) \\ & + \rho \sum_{(u,v,w) \in \mathcal{S}} \max(h + d(u,v) - d(u,w), 0) \\ & + \gamma \mathcal{L}_{AM}\end{aligned}$$

[0044]

[0045] 이때, $0 \leq m_u, n_i, h_u \leq 1$ 이며, 여기서 $\mathcal{D}$ 및 $\mathcal{S}$ 는 훈련 데이터(training data)에서 얻은 유효한 삼중 집합들(triple sets)이다. 가중치 λ, ρ, γ 는 그리드 서치에 의해 설정되는 하이퍼파라미터들(hyperparameters)이다. 각 미니 배치가 끝날 때마다 아래와 같이 모든 임베딩 벡터가 동시에 정규화되어 과적합을 방지할 수 있다.

$$z_a \rightarrow \frac{z_a}{\max(\|z_a\|, 1)} \text{ for all } a \in \mathcal{U} \cup \mathcal{V}$$

[0046]

[0047] 본 발명에서의 그래디언트 업데이트 과정은 도 1에 도시되어 있다. 유저이자 트러스터인 u 와, 트러스티 v , 연결되지 않은 유저를 w , 포지티브 아이템을 i , 네거티브 아이템을 j 로 표기하였다. 원 핫 벡터를 룩업 인덱스로 사용하여 유저와 아이템의 임베딩 벡터를 초기화한다. 유저 중심 손실, 아이템 중심 손실 및 사회 중심 손실인 세 가지 손실을 이용하여, 트레이닝 단계 t 에서의 각 임베딩은 훈련 단계 $t+1$ 로 표시된 대로 업데이트된다.

[0048]

다음은 u 에 대한 그래디언트 계산이다. 유저 u 는 v_i 들의 트러스터이고, w_j 는 유저 u 의 비연결 유저이며, N_u 는 유저 u 의 트러스티의 수이고, M_u 는 유저 u 의 비연결 유저의 수다. 첫 번째 그래디언트는 다음과 같이 정의된다.

$$d_{u1} = 2\rho \sum_{i=1}^{N_u} \sum_{j=1}^{M_u} (z_{w_j} - z_{v_i})$$

[0049]

[0050] a_i 들의 트러스티가 유저 u 이며, a_i 의 수는 K_u 이고, 두 번째 그래디언트는 다음과 같이 정의된다.

$$d_{u2} = 2\rho \sum_{i=1}^{K_u} (z_u - z_{a_i})$$

[0051]

[0052] b_i 들의 비연결 유저가 유저 u 이며, b_i 의 수는 K'_u 이고, 세 번째 그래디언트는 다음과 같이 정의된다.

$$d_{u3} = 2\rho \sum_{i=1}^{K'_u} (z_{b_i} - z_u)$$

[0053]

[0054] 최종적으로 그래디언트 업데이트는 다음과 같다.

$$\begin{aligned}z_u^{new} = & z_u^{old} \\ & - \eta (d_{u1} + d_{u2} + d_{u3}) \\ & + 2 \sum_{(u,i,j) \in \mathcal{D}} (z_j - z_i) \\ & + 2\lambda \sum_{(u,i,j) \in \mathcal{D}} (z_u - z_i)\end{aligned}$$

[0055]

[0056] 이때, η 는 학습률(learning rate)이다. 추가적으로, 개인화된 마진의 업데이트는 다음과 같다.

$$h_u^{new} = \max \left(h_u^{old} + \eta \frac{1}{|\mathcal{U}|}, 1 \right)$$

[0058] 그래디언트 업데이트에 관한 보다 구체적인 설명은 Li 등의 논문(Li et al., 2020)이 참조될 수 있다.

[0059] 복잡도 비교

[0060] 본 발명의 시간 복잡도는 $O(d(|\mathcal{D}| + |\mathcal{S}|))$ 이다. 기존의 알고리즘인 CML과 SML의 시간 복잡도는 각각 $O(d(|\mathcal{D}|))$ 와 $O(d(|\mathcal{D}|))$ 이다. 일반적으로, 실제 데이터의 아이템의 숫자 보다 유저의 수가 적기 때문에, $|\mathcal{D}| + |\mathcal{S}|$ 가 $2|\mathcal{D}|$ 보다 작다. 따라서, $O(d(|\mathcal{D}| + |\mathcal{S}|))$ 는 $O(d(|\mathcal{D}|))$ 에 근사한다. 크기가 k(작은 숫자)인 음수 샘플링이 적용되면 $|\mathcal{D}| = k|\mathcal{U}| \times |\mathcal{V}|$ 및 $O(d(|\mathcal{D}|)) = O(d(|\mathcal{U}| |\mathcal{V}|))$ 가 성립한다.

[0061] 코사인 유사도 행렬을 계산하는 시간 복잡도는 $|\mathcal{U}| |\mathcal{V}| |\mathcal{U}|$ 이므로, 코사인 유사도 행렬과 등급 예측을 위한 행렬 인수분해로 구성된 모델은 $O(|\mathcal{U}| |\mathcal{V}| |\mathcal{U}|) + d(|\mathcal{U}| + |\mathcal{V}|)$ 이다. 코사인 유사도 행렬을 계산하는 것은 시간 복잡도가 더 크기 때문에 $O(|\mathcal{U}| |\mathcal{V}| |\mathcal{U}|) + d(|\mathcal{U}| + |\mathcal{V}|) > O(d(|\mathcal{D}|))$ 이므로, 본 발명은 더 간단하고 가벼운 알고리즘임이 검증되었다고 말할 수 있다.

[0062] [실시예]

[0063] 본 발명에서 실험한 데이터 세트에는 Ciao 및 Yelp-Hotel(2020년 2월 21일에 마지막으로 업데이트된 데이터)이 포함되었다. 이러한 데이터 세트는 적절한 전처리 후 신뢰 네트워크가 있는 추천 시스템에서 널리 사용된다. Ciao 데이터 세트는 각 유저가 최소 5개의 상호 작용 항목을 가지고 있으며 각 아이템은 최소 5개의 상호 작용 사용자를 가지고 있다는 조건을 충족하기 위해 사전 처리되었다. 지정된 유저에 대한 공통 아이템이 없는 트러스터는 필터링할 수 있다.

[0064] Yelp-Hotel 데이터는 각 유저가 최소 10개의 상호 작용 항목을 가지고 있으며 특정 유저에 대한 공통 아이템이 없는 트러스터가 필터링되도록 사전 처리될 수 있다.

[0065] 그런 다음 사전 처리된 데이터로 트러스트 네트워크가 형성되었으며, 결과적으로 일부 사용자는 트러스트를 갖지 않을 수 있다. 사용자별로 상호작용 항목을 64:16:20의 비율로 학습(train), 검증(validation) 및 테스트(test) 세트로 나누어 실험을 진행하였다. 데이터의 크기 때문에 각 사용자 u에 대한 모든 네거티브(관계가 없는) 아이템을 나열하려면 종종 엄청난 계산 부하가 필요하다. 계산 부담을 줄이기 위해 499개의 네거티브 아이템과 499개의 비연결 사용자를 무작위로 선택하여 네거티브 샘플링을 적용할 수 있다.

[0066] 아래의 표 1과 표 2는 기존의 방법론들인 BPR, DASR, DBTS, LOCABAL, LOCABAL+와 본 발명의 성능을 비교한 결과이다. P@5, R@5, M@5, N@5, P@10, R@10, M@10, N@10 모두 높을수록 좋은 성능임을 감안할 때, 아래의 표에서 알 수 있듯이, 본 발명은 기존 방법론들보다 좋은 성능을 보임을 알 수 있다.

[0067] [표 1]

	P@5	R@5	M@5	N@5	P@10	R@10	M@10	N@10
BRP	0.0282	0.0277	0.0144	0.0357	0.025	0.0477	0.0086	0.0415
DASR	0.002	0.005	0.0001	0.0003	0.0015	0.0037	0.0002	0.0018
DBTS	0.0179	0.0241	0.0104	0.0274	0.0128	0.0342	0.0057	0.0295
LOCABAL	0.0033	0.0038	0.0018	0.0047	0.003	0.0069	0.0011	0.0056
본 발명	0.0322	0.0322	0.0169	0.0416	0.0259	0.0512	0.0097	0.0457

[0068]

[0069] [표 2]

	P@5	R@5	M@5	N@5	P@10	R@10	M@10	N@10
BRP	0.2965	0.556	0.1992	0.5812	0.1815	0.6514	0.104	0.6176
DASR	0.0009	0.0016	0.0005	0.0016	0.0012	0.0039	0.0004	0.0026
DBTS	0.0575	0.1169	0.0347	0.1102	0.039	0.1532	0.0187	0.1246
LOCABAL	0.0076	0.0107	0.003	0.0091	0.0096	0.0274	0.0022	0.0164
LOCABAL+	0.0075	0.0097	0.0026	0.0081	0.0105	0.0283	0.0022	0.0161
본 발명	0.4009	0.625	0.2635	0.6991	0.2265	0.6777	0.1352	0.7095

[0070]

[0071]

도 2는 본 발명의 일 실시예에 따른 메트릭 학습 방법을 설명하기 위한 흐름도이다. 메트릭 학습 방법은 사회적 협업 메트릭 학습 방법, 추천 방법, Top-K 추천 방법 등으로 명명될 수도 있다. 메트릭 학습 방법은 적어도 프로세서(processor) 및/또는 메모리(memory)를 포함하는 컴퓨팅 장치에 의해 수행될 수 있다. 따라서, 메트릭 학습 방법을 구성하는 단계들 중 적어도 일부는 컴퓨팅 장치에 포함되는 프로세서의 동작으로 이해될 수도 있다. 컴퓨팅 장치는 PC(Personal Computer), 서버(server), 태블릿 PC, 노트북 등을 포함할 수 있다. 이하에서, 메트릭 학습 방법을 설명함에 있어, 앞선 기재와 중복되는 내용에 관하여는 그 구체적인 기재를 생략하기로 한다.

[0072]

우선, 데이터가 획득된다(S110). 데이터는 복수의 유저들에 관한 정보(예컨대, 유저를 식별하기 위한 정보 등), 복수의 아이템들에 관한 정보(예컨대, 아이템을 식별하기 위한 정보 등), 유저들 각각의 아이템과의 관련성(또는 연관성으로, 해당 아이템에 대한 평가의 유무를 의미할 수 있음)과 평가의 정도(평점, 별점 등을 의미할 수 있음), 유저 간의 트러스터-트러스티 관계의 유무(사회적 네트워크를 의미할 수도 있음) 중 적어도 일부를 포함할 수 있다.

[0073]

유저들 및 아이템들 각각에 대한 임베딩 벡터가 생성된다(S120). 이에 따라, 유저 임베딩 행렬(user embedding matrix)과 아이템 임베딩 행렬(item embedding matrix)이 생성될 수 있다.

[0074]

벡터 공간(또는 메트릭 공간)에서의 손실이 계산될 수 있다(S130). 손실은 앞서 소개한 손실 함수를 이용하여 계산될 수 있다. 특히, 사회 중심 손실을 채용함으로써 보다 정확한 추천 아이템을 선별할 수 있다. 이때, 손실은 유저 중심 손실, 아이템 중심 손실, 사회 중심 손실, 및 적응형 마진 손실이 개별적으로 계산되어 가중합되거나, 또는 전체 손실 함수를 이용하여 전체 손실이 계산될 수 있다.

[0075]

임베딩 벡터에 대한 그라디언트 업데이트가 수행된다(S140). 그라디언트 업데이트로써, 유저 및 아이템에 대한 임베딩 벡터가 정규화되고 메트릭 공간에서 재배치될 수 있다.

[0076]

여기서, S130 단계와 S140 단계는 미리 정해진 시점(예컨대, 미리 정해진 시간 구간 동안, 또는 손실이 미리 정해진 값보다 작아질 때까지의 시점) 까지 반복적으로 수행될 수 있다.

[0077]

적어도 하나의 유저에 대하여, 아이템들과의 거리가 계산된다(S150). 여기서, 거리는 유클리드 거리를 의미할 수 있으나, 본 발명의 권리범위가 이에 제한되는 것은 아니다.

[0078]

산출된 거리를 기초로 적어도 하나의 유저에게 추천할 K(K는 임의의 자연수) 개의 아이템이 추출될 수 있다(S160). 추출되는 아이템은 가장 작은 거리를 나타내는 K 개의 아이템일 수 있다.

[0079]

상술한 과정을 거쳐 추출된 K 개의 아이템은 각 유저에게 추천될 수 있다.

[0080]

이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 컨트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로 컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.

[0081]

소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의

조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0082]

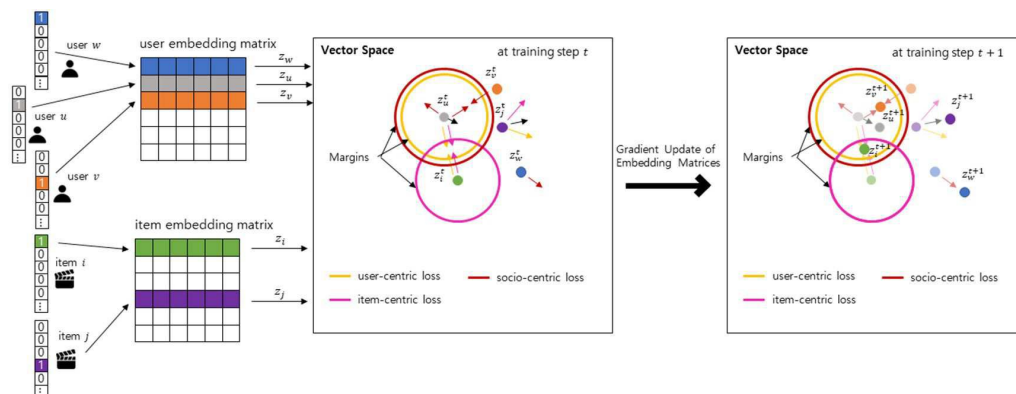
실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0083]

본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

도면1



도면2

