Structural Recommendations in Networks

이혜승

Contents

Introduction - What is Ranking Algorithms

Introduction - What is Ranking Algorithms

- 검색 기술도 추천시스템과 비슷하게 user에게 검색 결과로 콘텐츠를 추천.
- Traditionally, 방대한 양의 웹 유저들의 tracking 자체가 어렵기때문에,

  특정 user에게 개인화된 추천이 어려웠음.

- 최근에는 personal interest 에 집중한 웹페이지 추천의 개념이 생김.
- 네트워크에 노드 순위를 매기는 것과 동일.(ranking nodes)

Introduction - What is Ranking Algorithms

- 10.2.5 search & recommendation

  검색과 추천에는 서로 밀접한 관련 있음.
- 주요 차이점: **personalization.**

  검색할 때, 내 취향에 맞는 결과를 기대X. 검색어에 맞는 high quality의 검색 결과를 기대.

- 최근에는 검색과 추천의 문제가 통합됨.

  Ex.구글 검색결과는 user의 위치/검색 기록에 따라 달라짐(browser 설정 및 계정 로그인 상태에 따라)

Introduction - What is Ranking Algorithms

Different variations

- 1. recommending nodes by authority and context :

- Incoming link에 의해 노드의 품질을 판단.

: 하이퀄리티노드는 incoming links가 많음

- 개인화된 관련성은 그 문맥에 의해 판단.

검색엔진과 매우 밀접.

주요 observation: 다양한 사용자를 구별하지 못하기 때문에 특정 사용자에게 개인화되지는 않음.

서치엔진에서, 웹페이지(노드)들은 authority, context 기반으로 순위가 매겨짐.

검색을 하는 user의 identity 에는 전혀 중점x.

최근, personalized PageRank처럼 개인들의 다양한 관심사에 맞게 결과를 조절할 수 있는 알고리즘들도 등장.

Introduction - What is Ranking Algorithms

Different variations

- 2.recommending nodes by example: 많은 추천문제에 적용됨.

우리는 다른 example nodes와 **비슷한** nodes를 추천하고 싶어함.

이 문제는 nodes의 collective classification임.

Personalized PageRank 방법은 이 문제에서 잘 쓰임.

따라서, 이런 추천의 두가지 형태가 밀접한 연관이 있음.

이러한 적용은 user들과 다른 형태의 nodes들의 information network 에 유용하게 사용된다.

Introduction - What is Ranking Algorithms

Different variations

- 3.recommending nodes by influence and content:

영향력과 내용에 따라 노드 추천.

This problem은 viral marketing이라고 불림.

판매자는 자신의 특정 제품에 대해 propagate views할 user를 찾는다.

영향력 분석문제: viral 잠재력 + 그들의 주제적 특수성을 근거로.

Introduction - What is Ranking Algorithms

Different variations

- 4.recommending links:

소셜 네트워크에서는 네트워크의 접속성을 높이는 것이 SNS의 이익임.

따라서 잠재적인 친구 or 콘텐츠를 추천해 줌.

이 문제는 네트워크의 잠재적 링크 재조정하는 것과 동일함.

링크예측에는 많은 ranking methods가 존재. (Matrix factorization도 링크예측에 적용 가능)

또한, collective classification에 적용되기도 함.

[*PageRank*]

PageRank

- PageRank는 웹 서치의 quality를 높이기 위해서 제안된 알고리즘.

- GOOGLE의 공동창업자인 세르게이 브린과 래리 페이지가 처음 제안
  (The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin and Lawrence Page)
  현재는 더욱 정교한 알고리즘 but 기본 구조나 아이디어는 동일.

- 초기 서치 엔진들이 직면한 문제:
  contents-centric 웹페이지 및 순위 매칭 → 결과 품질 및 성능 저하.
  (∵ 인터넷에는 spam, 오해의 소지가 있는 정보, 부정확한 콘텐츠들이 존재 → 정보들의 품질을 구별해서 매칭할 수 없었음.)

PageRank

- 웹 페이지들의 reputation 과 quality 를 결정하는 매커니즘이 필요.

- **Citation** structure of the Web을 통해 가능해짐.
  기본 아이디어: 한 web page의 퀄리티가 좋다면, 다른 web page들이 해당 페이지를 가리킬 것(link).
  투표의 느낌으로 이해할 수 있음.

- PageRank 알고리즘은 더 전체론적인 citation based vote를 제공하여, 재귀적으로 랭킹을 일반화한다.

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank

- **노드의 중요성**을 모델링한다. (웹 그래프에서 linkage 개념을 활용)

- 기본 아이디어: 평가가 좋은 웹은 다른 좋은 웹에서 **인용**이 많이 되었을 것이다.

  - node = web page

  - edge = hyperlink

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank - Basic random surfer model

- (상황 설명) 페이지에서 임의의 링크를 선택해서 임의의 페이지에 방문하는 random surfer.
- 특정 페이지에의 long-term relative frequency of visits

  : in-linking pages의 숫자에 영향을 분명 받는다.

  즉, 해당 페이지로 들어오는 in-linking pages들이 많으면 당연히 유입도 높을 것
- 자주 방문하는 다른 페이지 or reputation 이 좋은 페이지와 연결된다면, 그 페이지에 대한 long-term relative frequency of visits는 높아질 것
- Ex) Stanford.edu와 조회수가 높은 페이지에서, 어떤 페이지 링크를 건다면 구글 검색 순위가 상승.

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank - Basic random surfer model

- (상황 설명) 페이지에서 임의의 링크를 선택해서 임의의 페이지에 방문하는 random surfer.

- **PageRank 알고리즘은 random surfer의 long-term relative frequency of visits관점에서 웹 페이지의 reputation을 모델링.**
- long-term relative frequency은 steady-state probability라고 하며,

  이 모델은 random-walk model라고 부른다.

PageRank

Basic한 random surfer 모델이 잘 작동하지않는 상황

- 1) dead-end:

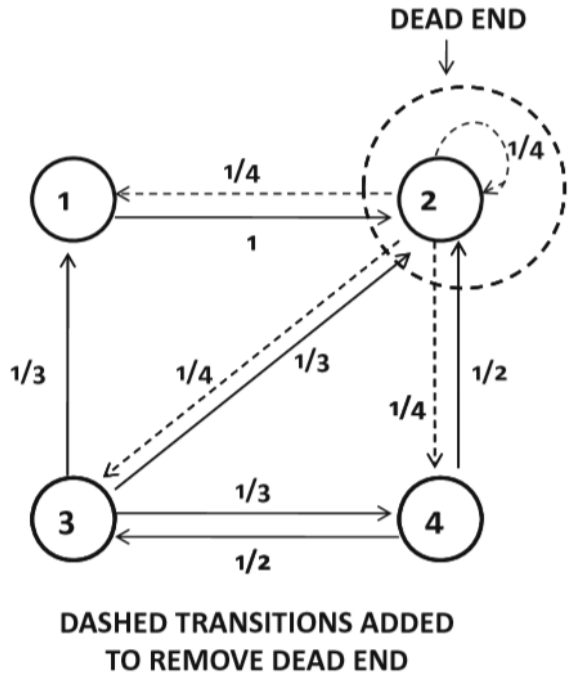
  random surfer 가 접속한 노드(page)에서 나가는 link가 없는 경우

  → 해당 웹 페이지에 random surfer가 갇혀서 다른 페이지로 이동이 불가능

- Solution

  dead-end 노드에서 자기 자신을 포함한 모든 노드로 links를 추가

  → 각 edge들은 1/N의 transition 확률을 갖게 된다.(N: 전체 노드 개수)



PageRank

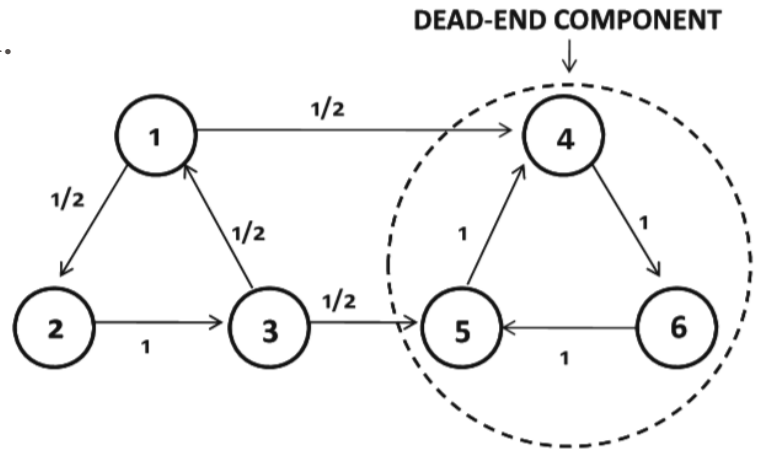
Basic한 random surfer 모델이 잘 작동하지않는 상황

- 2) dead-end component(absorbing component) :

  노드 그룹에서 다른 노드로 나가는 링크가 없는 경우, No cyclic graph.

  dead-end node problem에서 edges를 추가하면 해당 문제는 해결

  dead-end component problem 해결하려면 추가 단계가 필요!!



- Solution

  Random surfer모델 내에서 **restart or teleportation** 단계가 추가.

  각 transition 시, random surfer가 **확률 α의 임의의 페이지로 점프** or 확률 $1-\alpha$의 링크(edge)를 따름.

  α가 클 수록, 다른 페이지가 더 균등해질 수 있는 steady-state확률의 결과를 낸다. 즉, $\alpha = 1$, 모든 페이지에 접속할 확률은 동일.

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank - steady-state probabilities

- $In(i)$: node i에서 발생한 노드 집합(node i로 들어오는 노드 집합)
- $Out(i)$:node i에서 outgoing 노드 집합
- $\pi(i)$: node i의 steady-state 확률.
- 확률 p_{ij}= 1 / |Out(i)|

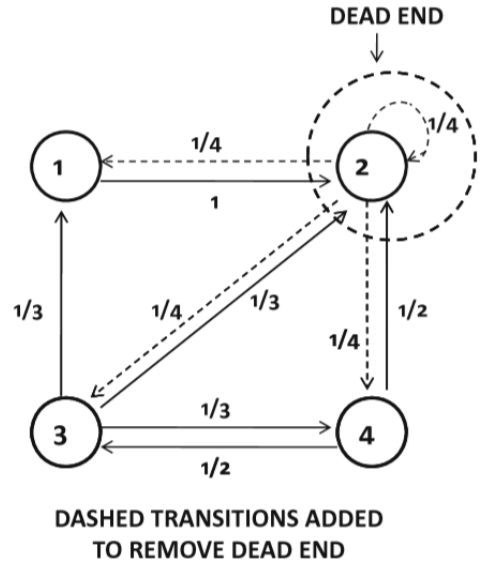
  : node i에서 node j로 전환(transition)할 확률
- node i의 PageRank = 'Markov chain model의 node i에 대한 steady-stat확률 $\pi(i)$'

$$\boldsymbol{\pi(i) = \frac{\alpha}{n} + (1-\alpha) \sum_{j \in in(i)} \pi(j)\, p_{ji}}$$

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank - steady-state probabilities

- $\pi(i) = \frac{\alpha}{n} + (1-\alpha)\sum_{j \in in(i)} \pi(j)\, p_{ji}$
- Ex) figure 10.1



$$\pi(2) = \frac{\alpha}{4} + (1-\alpha) * \left(\pi(1) + \frac{\pi(2)}{4} + \frac{\pi(3)}{3} + \frac{\pi(4)}{2}\right)$$

- 즉, 노드마다 이런 방정식이 하나씩 있을 것이므로, 매트릭스 형태로 나타낼 수 있음.

PageRank

일반적인 웹 랭킹의 맥락에서의 PageRank - steady-state probabilities

- $\bar{\pi} = (\pi(1) \cdots \pi(\mathrm{n}))^T$, $\bar{e}$: 1로 채워진 n차원 열 벡터.

$$\bar{\boldsymbol{\pi}} = \frac{\boldsymbol{\alpha}\bar{\boldsymbol{e}}}{\boldsymbol{n}} + (\mathbf{1} - \boldsymbol{\alpha})\, \boldsymbol{P}^{\boldsymbol{T}}\bar{\boldsymbol{\pi}}$$

- 알고리즘은 t=0일때부터 시작해서, 계속 $\bar{\boldsymbol{\pi}}^{(t+1)}$ 과 $\bar{\boldsymbol{\pi}}^{(t)}$의 차이가 임계값보다 작을 때까지 반복해서 $\bar{\boldsymbol{\pi}}^{(t+1)}$ update.
- 모든 노드 쌍 사이에 restart edge를 추가해서, transition matrix 에서 restart효과를 직접적으로 통합.

[*Personalized PageRank*]

Personalized PageRank

- PageRank: 링크 구조상 famous노드를 찾기에 적절 but 개인 관심사에 잘 맞는 item을 찾기에는 어려움.
- Personalized PageRank(= Topic-sensitive PageRank) : 개인화된 검색 결과 제공을 위해 등장.

- user가 다른 topics보다 특정 조합의 items에 관심있을 수 있음.
  Ex) 자동차에 관심있는 유저에게는 자동차 관련 페이지의 순서를 높게 지정해주고자 함.
  (personalization of ranking values.)

Personalized PageRank

- How to achieve?
- PageRank 방정식을 수정.

  이때. teleportation이 웹 문서의 전체 공간이 아니고, 해당 웹 문서의 샘플 세트에서만 수행.

- $\overline{e_p}$: 각 페이지마다 하나의 entry를 갖는 n차원 열 벡터.

  만약 그 페이지가 샘플 세트를 가지고 있으면 1, o.w 0.
- n_p: $\overline{e_p}$가 1인 개수.

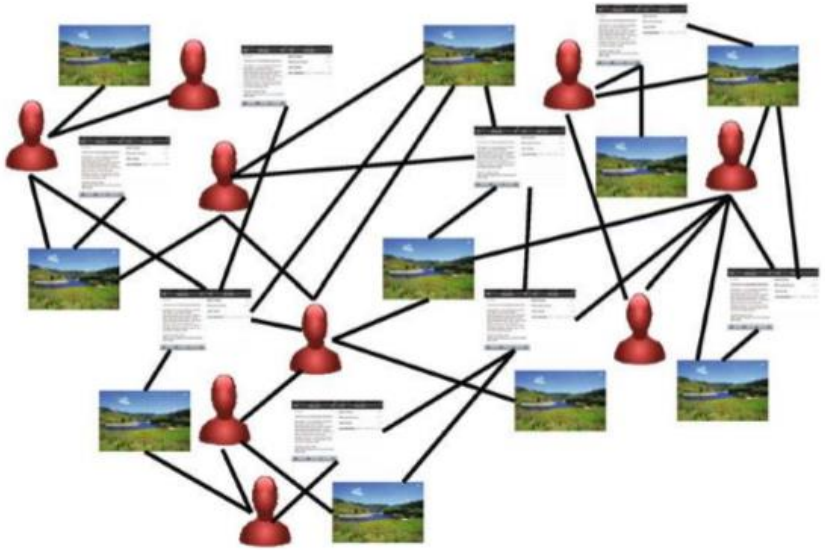
$$\bar{\pi} = \frac{\alpha \overline{e_p}}{n_p} + (1 - \alpha) P^T \bar{\pi}$$

→ 구조적으로 비슷한 locality에 있는 페이지가 더 높은 순위를 갖게 됨.

- α가 클수록 topic sensitive, 작을수록 네트워크 구조에 민감해짐.

Personalized PageRank

Personalization in Heterogeneous social media



- 동일한 network에 users, media content, text 설명 등의 다양한 양식의 노드를 갖는 형태.

- 네이버 bizAi: Line coupon 추천모델 개발 시,
  CF보다 정확도를 높이기 위해, graph기반 NN활용.
  user와 coupon 을 모두 노드로 보고, 쿠폰과 유저 간 interaction이 있다면 라인을 연결하는 모델.
  (+) 기존의 추천 방법으로는 기록이 적은 item과 user에 대해서는 추천을 잘 못해줬는데, 그래프 모델은 edge가 하나라도 존재하면, 모든 아이템과 유저간 정보가 전파(propagate)되어서 보다 쉽게 해결됨.
  즉, cold-start problem 해결.

- 참고할 만 한 논문: Tripartite Heterogeneous Graph Propagation for Large-scale Social Recommendation, https://arxiv.org/abs/1908.02569

[*SimRank*]

SimRank

- 기본 아이디어: 비슷한 페이지에 의해서 가리켜지면, 비슷한 페이지일 것이다
- 계산 과정은 PageRank와 유사함
- But 관점의 차이

  - PageRank: 해당 노드에서 가장 영향력이 큰 페이지를 찾음.

  - SimRank: 노드 간의 유사도를 측정해서 비슷한 페이지를 찾음.

- 그래프에서의 구조적인 특성(Structural feature) 만 고려
- (-) 각 user에서 공통 노드로 가는 경로가 같은 길이어야 함. 결과적으로 동일한 길이의 경로가 공통으로 존재하지않으면 직접 연결된 두 노드 사이의 simrank = 0이 될 수 있음.
- (-) 두 노드가 비슷하지만, odd lenght(홀수의 길이)를 가진 path만으로 접근 가능할 때는, 유사도가 떨어지게 나올 수 있음.

SimRank

- 기본 idea: 비슷한 object에 의해 참조되는 두 object는 비슷할 것.
  동일한 object의 SimRank에는 1 부여.
- SimRank는 노드들 사이의 대칭적인 유사도 결정. (i와 j의 유사성 = j와 i의 유사성)
- Basic SimRank equation

$$s(i,j) = \begin{cases} 1, & i = j \\ \dfrac{C}{|In(i)||In(j)|} \displaystyle\sum_{p \in In(i)} \sum_{q \in In(j)} SimRank(p,q), & i \neq j \end{cases}$$

- C: decay rate of the recursion
- 초기화: i=j일 때 1, 아니면 0 → 재귀적 반복.

[*Recommendations by Collective Classification*]

Recommendations by Collective Classification

- Collective classification: 추천과정에서 contents를 통합하는데에 효과적.
- Ex. 골프 제조업체는 골프에 관심있는 user들을 결정하고자 할 것. 제조사는 이미 골프에 관심있는 개인들의 사례 정보를 갖고 있을 것임(user profile, like 버튼 등)
- 네트워크에서 특정 행동의 카테고리는 label로 볼 수 있음. Label이 붙은 것은 트레이닝셋, 지정되지않은 것은 결정하고자하는 test셋.
- 기본적으로, 비슷한 이웃들을 기반으로 판단하고자 함.
- 1. 주어진 노드에 근접한 K개의 라벨링된 노드를 검사
- 2. k개 중 다수의 label을 해당 노드의 label로 배정.
- → sparsity때문에 사실 어려움.
- (해결 방법): 라벨링되지않은 노드를 통한 간접 연결도 필요하다. 1. 반복분류 / 랜덤 워크

참고 및 추천

- Tripartite Heterogeneous Graph Propagation for Large-scale Social Recommendation, https://arxiv.org/abs/1908.02569
- Graphs, Entities, and Step Mixture, https://arxiv.org/abs/2005.08485
- PageRank contents, https://sungmooncho.com/2012/08/26/pagerank/
- SimRank contents, https://frhyme.github.io/python-libs/nx_alg_simrank/
- 네이버 검색의 문제점과 발전 방향(추천), https://sungmooncho.com/2010/03/21/naver/

$$[\ Q\ n\ A\ ]$$