Week5-2

1. 학습정리
    1. 페이지랭크
        1. 들어오는 간선(투표)이 많을수록 신뢰
        2. 가중 투표
            1. 신뢰할 수 있는 웹사이트의 투표를 더 중요하게 간주
            2. 자신의 페이지 랭크 점수 / 나가는 이웃 수
        3. 임의보행관점
            1. 임의의 웹페이지 방문할 확률을 1/

2.1 페이지랭크의 정의: 투표 관점

측정하려는 웹페이지의 관련성 및 신뢰도를 페이지랭크 점수라고 부릅시다

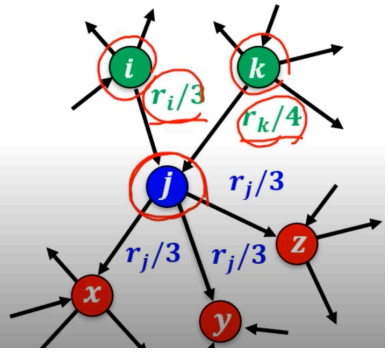
각 웹페이지의 페이지랭크 점수는 받은 투표의 가중치 합으로 정의됩니다

오른쪽 예시에서 웹페이지 j의 페이지랭크 점수 r_j는 다음과 같습니다

$$r_j = r_i / 3 + r_k / 4$$

페이지랭크 점수의 정의는 다음과 같습니다

$$r_j = \sum_{i \in N_{in}(j)} \frac{r_i}{d_{out}(i)}$$

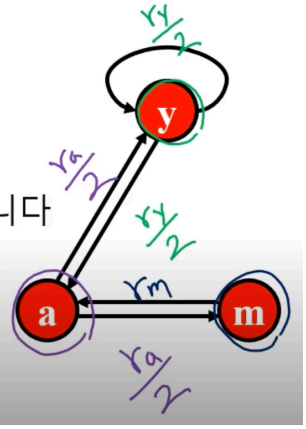


2.1 페이지랭크의 정의: 투표 관점

측정하려는 웹페이지의 관련성 및 신뢰도를 페이지랭크 점수라고 부릅시다

오른쪽 예시에서의 정점 별 페이지랭크 식은 다음과 같습니다

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$



변수 3개 식이 3개이므로 연립방정식을 통해 풀 수 있습니다
구체적인 계산은 뒤에서 자세히 설명합니다

2.2 페이지랭크의 정의: 임의 보행 관점

페이지랭크는 임의 보행(Random Walk)의 관점에서도 정의할 수 있습니다

임의 보행을 통해 웹을 서핑하는 웹서퍼를 가정합시다
즉, 웹서퍼는 현재 웹페이지에 있는 하이퍼링크 중 하나를
균일한 확률로 클릭하는 방식으로 웹을 서핑합니다

웹서퍼가 t번째 방문한 웹페이지가 웹페이지 i일 확률을 $p_i(t)$라고 합시다
그러면 $p(t)$는 길이가 웹페이지 수와 같은 확률분포 벡터가 됩니다

그러면 아래 식이 성립합니다

$$p_j(t+1) = \sum_{i \in N_{in}(j)} \frac{p_i(t)}{d_{out}(i)}$$

2. 페이지랭크 계산

3.1 페이지랭크의 계산: 반복곱

페이지랭크 점수의 계산에는 반복곱(Power Iteration)을 사용합니다

반복곱은 다음 세 단계로 구성됩니다

(1) 각 웹페이지 i의 페이지랭크 점수 $r_i^{(0)}$ 를 동일하게 $\frac{1}{\text{웹페이지의 수}}$ 로 초기화합니다

(2) 아래 식을 이용하여 각 웹페이지의 페이지랭크 점수를 갱신합니다

$$r_j^{(t+1)} = \sum_{i \in N_{in}(j)} \frac{r_i^{(t)}}{d_{out}(i)}$$

(3) 페이지랭크 점수가 수렴하였으면 종료합니다. 아닌 경우 (2)로 돌아갑니다

1. 반복곱이 항상 수렴하지는 않음
    1. 들어오는 간선은 있지만 나가는 간선은 없는 집합(스파이더 트랩) 이 존재하면 발생
2. 합리적인 점수로 수렴하는 것을 보장하지 않음
    1. 들어오는 간선만 있는 정점(막다른 정점)이 존재하면 발생
3. 순간이동
    1. 위 두 문제를 해결
        1. 스파이더 트랩과 막다른 정점에 갇힐 일 제거
        2.

3.2 문제점과 해결책

문제 해결을 위해 순간이동(Teleport)을 도입합니다

임의 보행 관점에서, 웹을 서핑하는 웹서퍼의 행동을 다음과 같이 수정합니다

√ (1) 현재 웹페이지에 하이퍼링크가 없다면, 임의의 웹페이지로 순간이동 합니다

(2) 현재 웹페이지에 하이퍼링크가 있다면, 앞면이 나올 확률이 α인 동전을 던집니다

(3) 앞면이라면, 하이퍼링크 중 하나를 균일한 확률로 선택해 클릭합니다

√ (4) 뒷면이라면, 임의의 웹페이지로 순간이동 합니다

(1)과 (4)의 임의의 웹페이지는 전체 웹페이지들 중에 하나를 균일확률로 선택합니다

순간이동에 의해서 스파이더 트랩이나 막다른 정점에 갇히는 일이 없어졌습니다

α를 감폭 비율(Damping Factor)이라고 부르며 값으로 보통 0.8 정도를 사용합니다

순간이동 도입은 페이지랭크 점수 계산을 다음과 같이 바꿉니다

(1) 각 막다른 정점에서 (자신을 포함) 모든 다른 정점으로 가는 간선을 추가합니다

(2) 아래 수식을 사용하여 반복곱을 수행합니다

$$r_j = \sum_{i \in N_{in}(j)} \left(\alpha \frac{r_i}{d_{out}(i)} \right) + (1-\alpha)\frac{1}{|V|}$$

$|V|$는 전체 웹페이지의 수를 의미합니다
파란색 부분은 하이퍼링크를 따라 정점 j에 도착할 확률을 의미합니다
빨간색 부분은 순간이동을 통해 정점 j에 도착할 확률을 의미합니다

- 의사결정 기반 전파 모형
  - 주변의 의사결정을 고려하여 각자 의사 결정
- 선형 임계치 모형
  - 가장 간단한 의사결정 기반 전파 모델

2.2 선형 임계치 모형

이 모형을 선형 임계치 모형(Linear Threshold Model)이라고 합니다

각 정점은 이웃 중 A를 선택한 비율이 임계치 q를 넘을 때만 A를 선택합니다

이 모형은 전부 B를 사용하는 상황을 가정합니다
그리고 처음 A를 사용하는 얼리 어답터들을 가정합니다
시드 집합(Seed Set)이라고 불리는 얼리 어답터들은 항상 A를 고수한다고 가정합시다

- 확률적 전파 모형
  - 독립 전파 모형
- 바이럴 마케팅
  - 전파 최대화
    - 영향력 있는 시드 집합 찾기
    - np-hard
    - 따라서 휴리스틱이나 근사 알고리즘 사용
- 휴리스틱
  - 실험적으로 잘 동작할 수 있지만 이론적으로 정확도를 보장할 수 없음
- 정점 중심성 휴리스틱
- 전파 최대화
  - 그리디 알고리즘
    - 최초 전파자를 한 번에 한 명씩 선택
    - 위 과정을 반복하며 전파자를 목표 수 만큼 늘림
    - 최초 전파자간의 조합의 효과를 고려하지 않음
    - 전파 최대화 하는 시드집합 반환을 보장하지 않음
    - 독립전파모형에서 정확도가 일부 보장

–

2. 피어세션
    1. 강의 리뷰
    2. 퍼더 퀘스쳔
        1. 감염병의 전파를 모델링 하기 위해서는 감염과 더불어 감염된 환자의 회복을 고려해야 합니다. 감염된 사람이 회복될 수 있고, 한 번 감염이 된 사람은 추가적으로 감염이 되지 않는다고 가정한다면 어떻게 전파 모델을 설계할 수 있을까요?
    3. 알고리즘 문제 풀이
        1. https://leetcode.com/problems/minimum-number-of-vertices-to-reach-all-nodes/
        2. https://leetcode.com/problems/is-graph-bipartite/submissions/
    4.