Глубинное обучение в анализе графовых данных

4. PageRank

в предыдущих сериях…

Эмбеддинги

1. перевод в пространство эмбеддингов
2. парадигма encoder-decoder
3. различные подходы к созданию эмбеддингов
    - inner-product methods
    - laplacian eigenmaps
    - random walks
    - node2vec
4. эмбеддинги для графов

Немного про свойства декодеров

Графы знаний (или multi-relational graphs)

Граф - $G = (V, E)$, V - вершины, E - ребра

В обычном графе $e = (u, v)$

В графе знаний $e = (u, t, v)$

В общем случае на графах знаний решается задача предсказания пропущенных связей, но бывают и задачи классификации вершин

Свойства

PageRank

PageRank

- это алгоритм, используемый для ранжирования страниц в интернете

Для сетевого анализа это наглядный пример использования всех полученных ранее знаний для создания готового способа решения насущной проблемы поиска

Что есть страницы в интернете

Будем рассматривать интернет в простом приближении (как это было на заре)

В интернете есть страницы (pages)

На страницах есть ссылки на другие страницы

Стоит **задача** - понять, какие страницы самые важные

Связь с графами

В таком простом приближении можно удобно представить интернет как огромный граф.

Вершины - страницы

Ребра - ссылки

Соответственно граф наш будет ориентированный - ребро между двумя вершинами будет направленным, оно будет означать что на странице А есть ссылка на страницу Б

Особенности такого графа

- Граф будет громадный
- Могут быть петли (на странице есть ссылка на саму себя)
- Ребра могут быть направлены из А в Б и из Б в А в одно и то же время (пример - главная страница ведет на подстраницу, на которой есть опция прыгнуть на главную страницу)

Интуиция

Самая очевидная идея - страница важна, если у нее очень много ссылок

Сразу возникает вопрос - какие ссылки важнее, **исходящие** или **входящие**?

Более того, возникает еще вопрос - а все ли ссылки **одинаково** важны?

Интуиция. Модель

Ссылка от важной страницы должна сигнализировать о важности страницы

Опишем модель:

- Вклад каждой ссылки должен быть пропорционален важности страницы, от которой она исходит
- Если страница i с важностью r_i имеет d_i ссылок, то каждая исходящая ссылка будет давать вклад r_i/d_i
- Страница j будет иметь важность, равную сумме входящих ссылок

рекурсивненько как-то…

Ранг

Как решать? Гауссов метод в случае многих миллиардов страниц будет не очень хорош

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

Матричная форма

Стохастическая матрица смежности:

Если есть ссылка из i в j, то $M_{ij} = 1/d_i$

Тогда переписать уравнения можно в матричном виде

$$r = M\, r$$

Интуиция за матричной формой

Попробуем понять, в чем смысл r = M r

Представим какого-то случайного пользователя в сети Интернет:

1. В момент времени t он попадает на страницу i
2. В момент времени t+1 он следует по случайной ссылке со страницы i
3. Он оказывается на странице j, попав на нее из i (**важно: j может быть страницей i)**
4. Процесс продолжается бесконечно

Пусть p(t) - вектор, у которого на i-том месте будет вероятность того, что юзер окажется на странице i в момент времени t

Тогда p(t) - вероятностное распределение на всех страницах

Интуиция за матричной формой

Движение по ссылкам случайным равновероятностным образом можно описать как $p(t+1) = M\, p(t)$

Представим, что в какой-то момент $p(t+1) = M\, p(t) = p(t)$

Тогда $p(t)$ - стационарное распределение случайного блуждания

Вспоминая $r = Mr$ получим, что r - стационарное распределение

Более того…

Вспомним про центральность через собственное значение

$$\lambda c = Ac$$

λ - собственное значение, c - собственный вектор

У нас r = Mr

Тогда подставим λ = 1, получим 1 r = M r

Вывод

Объединив все три идеи получим вывод -

r - собственный вектор стохастической матрицы смежности M с собственным значением 1

Начиная с любого вектора u M(M(...(Mu)) - долгосрочное распределение блуждающих юзеров

PageRank = Ограничивающее распределение = Главный собственный вектор M

Теперь можно решить задачу.

Решение PageRank

1. В начале назначим каждой вершине начальный ранг
2. Продолжать до сходимости (норма разницы между рангами в моменты t+1 и t меньше некого эпсилон)

Power Iteration

- алгоритм для решения задачи

1. инициализация - $r^0 = [1/N,...,1/N]$
2. шаг $r^{t+1} = M r^t$
3. повторять, пока не $|r^{t+1} - r^t|_1 < \varepsilon$
    a. $r := r^{t+1}$

Проблемы?

Какие вопросы возникают к PageRank?

- тупики (ломают)
- циклы (сходимость будет, но результаты не совсем те)

Уход от циклов

С помощью вероятностей можно научиться выбираться из цикла

- На каждом шаге с вероятностью \beta юзер выбирает одну из d_i ссылок на странице
- С вероятностью 1 - \beta юзер телепортируется

За конечное число шагов из цикла юзер выпрыгнет

Уход от тупиков

- заранее договориться, что в тупике сработает случайный телепорт

PageRank

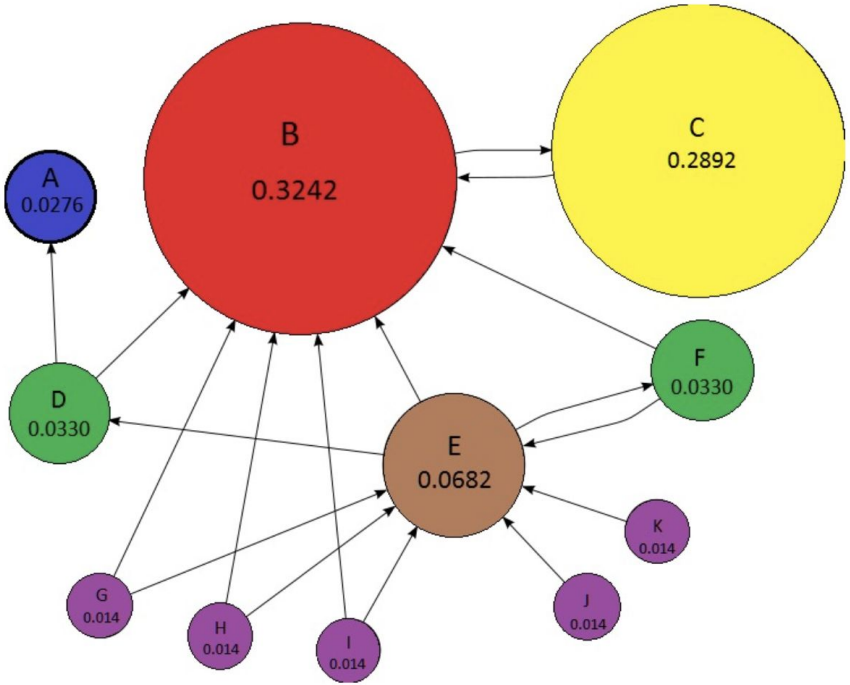
$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}$$

Новая матрица

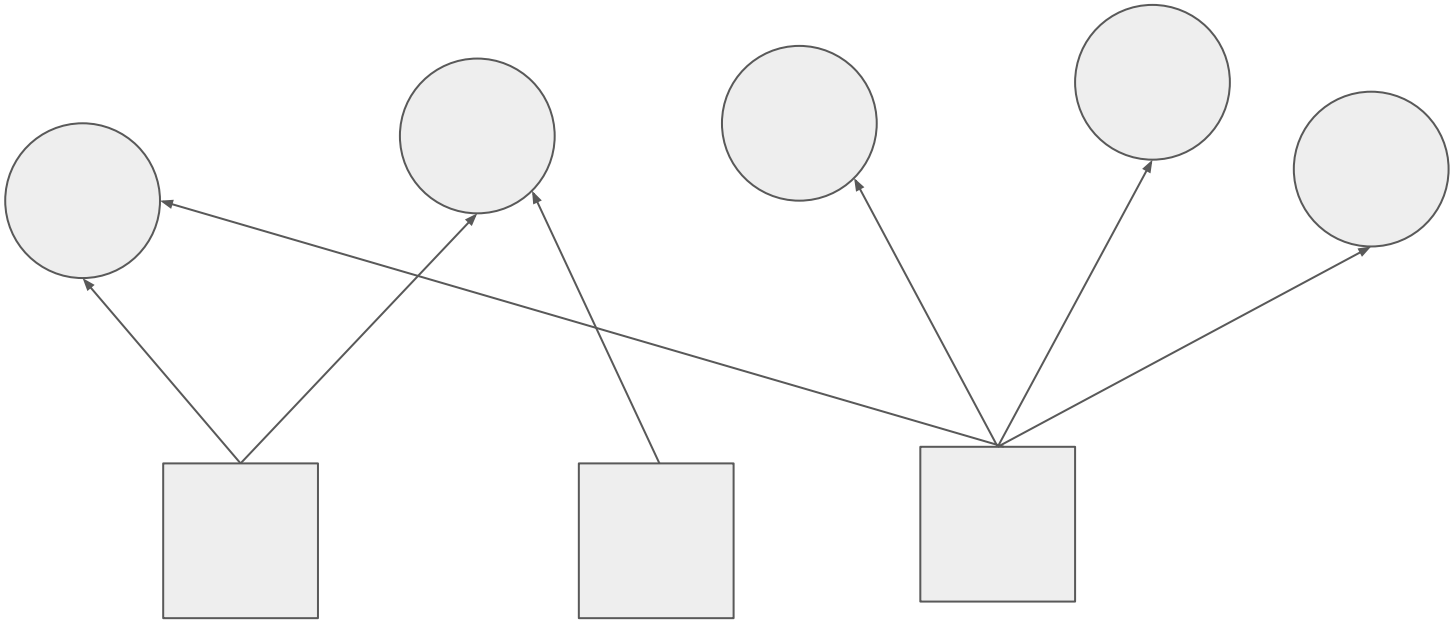
$$G = \beta\, M + (1 - \beta) \left[\frac{1}{N}\right]_{N \times N}$$

И получаем задачу r =Gr

PageRank пример



Рекомендации



Приложение к рекомендациям

- Меняем формат телепортации, ограничивая множество вершин, в которые будет совершен обратный прыжок
- Максимально похожий на какой-то элемент можно найти, ограничив множество для телепортации до одной конкретной вершины