

# Использование случайных блужданий по графу для повышения качества рекомендаций

Павел Коваленко

ВМК МГУ, 617 группа

Конференция «Ломоносов-2019»

Имеется множество пользователей  $U$  и множество объектов  $I$ . Для каждого пользователя  $u$  известен список объектов  $I(u)$ , которые он оценил.

Требуется для пользователя составить список из  $N$  объектов, которые он вероятно оценит в будущем.

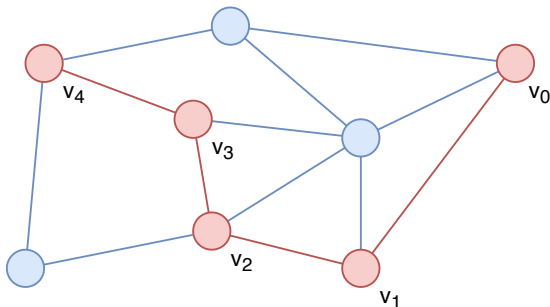
Каждому пользователю сопоставим вектор  $p_u \in \mathbb{R}^d$ , каждому объекту — вектор  $q_i \in \mathbb{R}^d$ .

Будем считать, что все объекты, которые пользователь не оценил, ему не релевантны.

Найдем такие вектора, чтобы для пользователя объекты были правильно упорядочены: если  $i \in I(u)$ ,  $j \notin I(u)$ , то  $\langle p_u, q_i \rangle > \langle p_u, q_j \rangle$ .

Теперь для пользователя  $u$  надо найти такие  $N$  объектов  $i$ , что  $\langle p_u, q_{i_k} \rangle$  максимально.

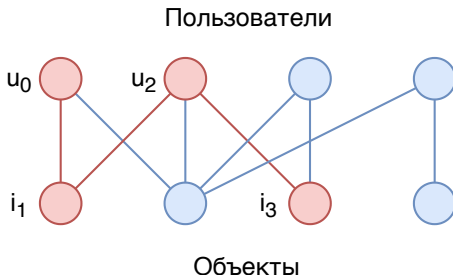
Случайное блуждание — это путь в графе, который задается следующим образом. Начальная вершина  $v_0$  выбирается случайно среди всех вершин графа. На последующих шагах новая вершина  $v_k$  выбирается случайно среди соседей вершины  $v_{k-1}$ . Также необходим критерий остановки, например, ограничение на длину пути.



## Случайные блуждания в задаче рекомендаций

Можно представить входные данные в виде двудольного графа. Вершины — пользователи и объекты, между пользователем и объектом есть ребро, если пользователь оценил объект.

Для составления рекомендаций для пользователя несколько раз начинаем случайные блуждания от этого пользователя и выбираем  $N$  вершин, которые в результате были посещены чаще других.



Обычно каждый пользователь оценил далеко не все объекты, поэтому матрица оценок оказывается разреженной и латентные представления плохо обучаются.

Авторы статьи *HOP-Rec: High-Order Proximity for Implicit Recommendation* предлагают использовать гибридную схему. Они обучают латентные представления стохастически, шаг обучения следующий:

- Случайно выбрать пользователя.
- Выбрать отрицательный пример случайно из всех объектов.
- Запустить случайное блуждание нечетной длины из пользователя и использовать вершину (объект), в которой оно остановится, в качестве положительного примера.

Таким образом искусственно создаются положительные примеры.

Описанный алгоритм оставляет много простора для выбора параметров случайных блужданий:

- Стартовая вершина — блуждания можно начинать с пользователя или с объекта
- Количество семплированных случайных блужданий
- Вероятность перехода в вершину. Обычно зависит от количества исходящих из вершины ребер:  $p(v) \sim \deg(v)^K$
- Критерий остановки
  - Блуждания фиксированной длины. Параметр — длина пути
  - Блуждания с фиксированной вероятностью остановки после каждого перехода. Параметр — вероятность остановки

Предложенный метод был испытан на датасетах MovieLens-[100K, 1M, 10M, 100M], Million Playlist и Amazon Review.

Испытанные алгоритмы — Alternating Least Squares, Bayesian Personalized Ranking и SVD++.

Метрики качества — Precision at 10, Normalized Discounted Cumulative Gain.



Датасет	P@10 исходный	P@10 модиф.	NDCG исходный	NDCG модиф.
MovieLens 100K	0.1098	<b>0.1195</b>	0.1173	<b>0.1208</b>
MovieLens 1M	0.1345	<b>0.1942</b>	0.1348	<b>0.1913</b>
MovieLens 10M	0.1496	<b>0.1788</b>	0.1488	<b>0.1721</b>
MovieLens 20M	0.1514	<b>0.1875</b>	0.1461	<b>0.1704</b>
Amazon Review	0.0125	<b>0.0145</b>	0.0119	<b>0.0132</b>
Million Playlist	0.1842	<b>0.2196</b>	0.1821	<b>0.2121</b>

Лучшая точность на рассмотренных датасетах для алгоритма  
Bayesian Personalized Ranking

Для большей части датасетов и алгоритмов удалось получить прирост точности рекомендаций за счет использования предложенного метода.

Разница наиболее заметна для Bayesian Personalized Ranking.

Лучшие параметры:

- Блуждания небольшой длины
- Вероятность перехода не зависит от степени конечной вершины
- Чем больше семплированных путей, тем выше точность