Основная идея

Возьмем за основу нейросетевой рекоммендер, так как он лучше всего показал себя в предыдущих экспериментах.

Добавим следующие улучшения:

1. В качестве контекста, помимо трека, с которого начинается сессия, будем использовать id пользователя. Гипотеза в том, что разные пользователи будут иметь разные предпочтения, даже если они начинают свою сессию с одного и того же трека.
2. Добавим к информации о текущем треке id исполнителя.
3. Усложним архитектуру сети в надежде на то, что она выучит более сложные взаимосвязи. К каждой из 2 веток (контекст, текущий трек) добавим «башню» из 3 полносвязных слоев (Linear + LeakyReLu). В конце будем считать скалярное произведение между выходами из каждой ветки. В качестве функции потерь так же будем использовать MSE.

Детали

Алгоритм получения данных для рекомендаций:

1. Собираем данные с помощью нейросетевого рекомендера (100000 сессий).
    a. Был проведен также эксперимент с обучением на данных рандомного рекоммендера, но в этом случае время прослушивания треков имеет неудачное распределение, сконцентрированное около 0, поэтому модель хуже показывает себя на реальных данных.
2. Выполняем обучение модели, используя early stopping.
3. Сохраняем эмбеддинги для треков, чтобы в дальнейшем использовать их в скалярном произведении, имея определенный контекст (пользователь и стартовый трек).
4. В сервисе Botify на старте приложения загружаем модель и эмбеддинги треков.
5. На каждый запрос для получения рекомендации
    a. Если это первый запрос в рамках текущей сессии, рассчитываем рекомендации для текущего пользователя и первого трека в сессии, сохраняем в Redis.
    b. Если это не первый запрос в рамках текущей сессии, берем рекомендации для пользователя и первого трека в сессии из Redis.
    c. Если это последний запрос в сессии, удаляем сессионные данные из Redis.

Код для обучения модели находится в файле recsys/jupyter/Recommender.ipynb.

Результаты A/B эксперимента

	treatment	metric	effect	upper	lower	control_mean	treatment_mean	significant
0	T1	time	32.913	42.891	22.936	5.506	7.318	True
1	T1	sessions	-1.330	1.349	-4.008	1.110	1.095	False
2	T1	mean_request_latency	242.525	254.242	230.807	0.683	2.341	True
3	T1	mean_tracks_per_session	16.487	21.351	11.623	9.968	11.611	True
4	T1	mean_time_per_session	34.882	44.425	25.340	4.961	6.691	True

Таблица1. Результаты сравнения нового рекоммендера с нейросетевым рекоммендером с помощью A/B эксперимента

Как видно из Таблицы 1, новый рекоммендер статистически значимо выигрывает у предыдущего рекоммендера (среднее время сессии и среднее количество треков в сессии увеличились). Также можно заметить, что увеличилось время ответа за счет того, что для выдачи рекомендаций теперь используется не статический список, а модель, учитывающая контекст запроса.