

9.4. Хорошие свойства рекомендательных систем

Введение

Если у нашей рекомендательной системы хорошие метрики ранжирования, то является ли она хорошей? При создании рекомендательной системы есть много рисков, например, создать кликбейт. Всё зависит от контекста, однако поговорим о наиболее распространённых критериях.

Полнота (Coverage)

Доля рекомендованных объектов среди всех объектов: $Coverage = \frac{|I_{recommended}|}{|I|}$

Данную метрику нужно рассматривать за разные временные интервалы(день, неделя), принимая во внимание возможные ограничения с объёмом данных. В случае музыкального сервиса скорее всего нужно будет заново рекомендовать наиболее понравившиеся треки, а в рекомендациях фильмов стоит это делать гораздо реже, то есть в рекомендациях фильмов полнота, скорее всего, будет расти быстрее.

На полноту влияет холодный старт. Часто на этапе холодного старта пользователям показывают наиболее популярные товары, из-за чего недавно добавленным товарам будет сложнее попасть к пользователю. Поэтому можно бустить свежие товары в течении какого-то времени.

Актуальные вопросы:

- Сколько нужно дней, чтобы полнота достигла заданного значения?
- Можно ли в принципе достигнуть этого значения с помощью нашего алгоритма?

Факторы, которые помогут ответить на эти вопросы:

- Какой объём трафика у нашей рекомендательной системы?
- Есть ли у бизнеса ограничения, влияющие на конечный список рекомендаций?
- Имеет ли наш алгоритм достаточную степень персонализации?
- Можно ли регулировать режимы exploration и exploitation во время работы рекомендательной системы?

Новизна (Novelty)

Один из способов оценить новизну - использовать статистическую меру собственной информации объекта (self information). Значение собственной информации для события X равняется логарифму вероятности наступления данного события. Согласно теории, чем меньше вероятность наступления события, тем больше потенциальной информации принесет это событие при его наступлении. Единицей информации при использовании логарифма по основанию 2 является бит. Чем менее популярен объект, тем более вероятно, что он будет новым для пользователя, то есть мера информации для такого объекта будет выше.

Для каждого рекомендованного объекта i считаем вероятность, с которой его порекомендуют случайному пользователю: $P_i = \frac{m_i}{N}$, где m_i – количество пользователей, которым был показан i -й объект, а N – общее число пользователей. Для заданного пользователя усредняем значение собственной информации по списку его рекомендаций R и получаем итоговое значение метрики: $Novelty_{user} = \frac{1}{|R|} \sum_{i \in R} -\log(P(i))$

Разнообразие (Diversity)

Разнообразие – это способность модели рекомендовать разные по содержанию объекты. Разнообразие можно рассчитывать на основе комбинаций метрик полноты и новизны. Также мерой разнообразия может быть дисперсия рекомендаций за заданный промежуток времени.

Помимо этого популярны подходы, использующие эмбединги объектов для оценки попарной схожести объектов и расчёта на основе неё значения разнообразия. Одна из таких метрик – Intra List Similarity (ILS). Чтобы ее посчитать, нужно иметь эмбединги объектов рекомендаций, находящиеся в едином векторном пространстве. Для расчёта разнообразия для одного пользователя нужно усреднить попарную схожесть sim между рекомендованными объектами:

$$ILS_{user} = \frac{1}{R} \sum_{i \in R} \sum_{j \in R} \text{sim}(i, j),$$

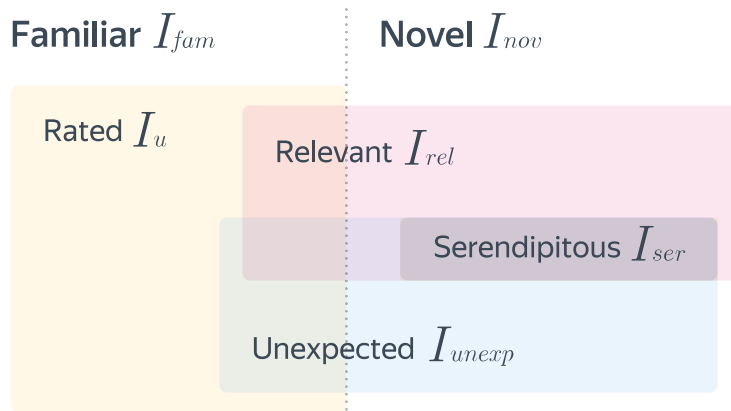
где R – это набор рекомендованных пользователю объектов.

Serendipity

На русский не переводится, что-то в духе "интуитивная прозорливость".

Serendipity – это способность рекомендовать такие объекты, которые не только релевантны для пользователя, но ещё и существенно отличаются от того, с какими

объектами пользователь взаимодействовал в прошлом.



Serendipity – довольно субъективное свойство и его сложно формализовать. Это свойство встречается редко, нет общепринятого способа подсчёта.

Один из способов: пусть R_u - список рекомендаций для пользователя, $Pr_u(i)$ - предсказание модели для каждого объекта из списка, а $Prim_u(i)$ - предсказание примитивной модели, а rel - известная релевантность объекта для пользователя, тогда:

$$Serendipity_{user} = \sum_{i \in R} \max(Pr_u(i) - Prim_u(i), 0) \cdot rel_u(i)$$

Ключевая идея формулы такова: если уверенность персонализированной модели в том, что пользователю понравится i -ый айтем, больше, чем уверенность неперсональной модели (примитивной), это значит, что данному пользователю может особенно понравиться i -й айтем.

Чтобы улучшить Serendipity:

- Добавлять больше фичей для пар пользователь-объект
- Взвешивать таргеты, чтобы наиболее точно учитывать взаимодействие
- Писать кастомные функции потерь

Кроме того, имеет смысл оптимизировать модель по метрике serendipity на офлайн тестовой выборке.

Заключение

Полезно смотреть на несколько метрик одновременно, чтобы оценить разные свойства моделей. На начальном этапе стоит концентрироваться на более простых и интуитивно понятных с точки зрения бизнеса метриках: конверсии, среднем времени визита и так

далее. А вот как только базовые метрики будут на удовлетворительном уровне, стоит начинать мониторить и оптимизировать метрики, разобранные в этом разделе.