Лекция 20
Рекомендательные системы

Е. А. Соколов
ФКН ВШЭ

26 июня 2021 г.

1 Метрики качества рекомендаций

Существует достаточно много метрик качества рекомендательных систем — некоторые связаны с точностью предсказания, а некоторые оценивают продуктовые аспекты (например, среднее качество тех фильмов, которые обычно рекомендуются). Нет общих советов по поводу того, на какую метрику имеет смысл обращать внимание. Один из возможных подходов — выбрать ключевую с точки зрения бизнеса онлайн-метрику (например, среднее время, которое пользователь проводит на сайте интернет-магазина, средний чек или что-то ещё), а затем выбрать оффлайн-метрику или линейную комбинацию оффлайн-метрик, которая лучше всего коррелирует с ключевой метрикой. Здесь под онлайн-метрикой понимается показатель, который можно измерить только при запуске рекомендательной системы на реальных пользователях, а под оффлайн-метрикой — функцию, которую можно оценить, построив предсказания модели на исторических данных. Также иногда пытаются найти промежуточную онлайн-метрику, которая коррелируют с основной, но при этом быстрее реагирует на изменения в работе рекомендательной системы — но эту тему мы пока не будем затрагивать. Разберём несколько оффлайн-метрик.

§1.1 Качество предсказаний

Поскольку рекомендательная система обучается предсказывать оценки r_{ui}, логично оценивать качество решения именно этой задачи.

Предсказание рейтингов. Если модель предсказывает рейтинг или другую вещественную величину (например, длительность просмотра), то качество может измеряться через MSE, RMSE, MAE или другие регрессионные метрики.

Предсказание событий. Если модель предсказывает вероятность некоторого события (клика, покупки, просмотра, добавления в корзину), то качество можно измерять с помощью метрик качества классификации — доля правильных ответов, точность, полнота, F-мера, AUC-ROC, AUC-PR, log-loss и т.д.

Также можно учитывать, что мы показываем пользователю только k товаров, получивших самые высокие предсказания модели, и нас интересует лишь качество

этих товаров. Если через $R_u(k)$ обозначить лучшие k товаров для пользователя u с точки зрения модели, а через L_u товары, для которых действительно произошло интересующее нас событие, то можно ввести следующие метрики:

- Наличие верной рекомендации: hitrate@k $= [R_u(k) \cap L_u \neq \emptyset]$;
- Точность: precision@k $= \frac{|R_u(k) \cap L_u|}{|R_u(k)|}$;
- Полнота: recall@k $= \frac{|R_u(k) \cap L_u|}{|L_u|}$.

Качество ранжирования. Вообще говоря, нам не очень важно, насколько точно модель предсказывает рейтинг или вероятность клика — от неё лишь требуется дать более релевантным товарам более высокие предсказания. Это значит, что модель должна правильно ранжировать (или сортировать) товары.

Одной из популярных метрик качества ранжирования является nDCG. Обозначим через a_{ui} предсказание модели для пользователя u и товара i. Отсортируем все товары по убыванию предсказания a_{ui}. Тогда для товара i_p на позиции p можно вычислить его полезность $g(r_{ui_p})$ и штраф за позицию $d(p)$. Метрика DCG задаётся как

$$\text{DCG@k}(u) = \sum_{p=1}^{k} g(r_{ui_p}) d(p).$$

Примерами конкретных функций могут служить $g(r) = 2^r - 1$ и $d(p) = \frac{1}{\log(p+1)}$. Чтобы значение метрики легче было интерпретировать, её можно поделить на значение DCG при идеальном ранжировании — в этом случае получим метрику nDCG (normalized DCG):

$$\text{nDCG@k}(u) = \frac{\text{DCG@k}(u)}{\max \text{DCG@k}(u)}.$$

Далее значение nDCG можно усреднить по всем пользователям.

Недостатки оценок качества предсказания. Основная проблема состоит в том, что качество предсказания само по себе не определяет пользу рекомендательной системы. Модель может идеально угадывать то, что купил пользователь — но, возможно, он приобрёл бы эти товары и без рекомендаций. Поскольку мы никогда не можем узнать, повлияли ли рекомендации на намерения пользователя, имеет смысл анализировать и другие метрики качества, которые могут косвенно говорить о пользе предсказаний модели [5].

§1.2 Покрытие

Покрытие товаров. Полезно обращать внимание на то, какая доля товаров в принципе рекомендуется пользователям — так, может оказаться, что модель показывает только самые популярные товары, а большая часть ассортимента игнорируется. В качестве простейшей метрики можно использовать *покрытие каталога*, которые вычисляется как доля товаров, порекомендованных хотя бы один раз.

Также можно оценить общее разнообразие рекомендаций. Пусть $p(i)$ — доля показа товара $i \in I$ среди всех показов для данной рекомендательной системы. Тогда разнообразие можно определить как энтропию такого распределения:

$$H(p) = -\sum_{i \in I} p(i) \log p(i).$$

Покрытие пользователей. Рекомендательная система может быть устроена так, что некоторым пользователям вообще ничего не рекомендуется — например, из-за низкой уверенности классификаторов или отсутствия тех или иных признаков для модели. Имеет смысл вычислять долю пользователей, для которых не рекомендуется ни одного товара, чтобы отслеживать проблемы с покрытием в модели рекомендаций.

§1.3 Новизна

Под новизной понимается доля новых для пользователя товаров среди рекомендованных. При этом под новыми понимаются те товары, которые пользователь видит впервые глобально, а не только на нашем сайте — в идеале хочется уметь угадывать, какие товары пользователь встречал раньше на других ресурсах.

Можно предложить несколько подходов к измерению новизны:

- Для каждого рекомендованного товара добавить в интерфейсе возможность сообщить о том, что этот товар пользователь уже видел.
- Удалить из обучающей выборки часть товаров, которые пользователь купил или просмотрел — тем самым мы будем моделировать ситуацию, в которой пользователь когда-то раньше узнал про этот товар, но в наших данных это не отражено. Далее будем оценивать новизну на основе того, как часто эти удалённые товары попадают в рекомендации.
- Можно считать, что пользователь с большей вероятностью встречал раньше популярные товары и с меньшей — непопулярные. Тогда новизну можно вычислять как долю угаданных рекомендательной системой товаров, где каждый товар имеет вес, обратно пропорциональный популярности этого товара.

§1.4 Прозорливость (serendipity)

Под прозорливостью понимается способность рекомендательной системы предлагать товары, которые отличаются от всех купленных пользователем ранее. Например, если пользователь читал только книги конкретного автора, то рекомендацию хорошей с точки зрения пользователя книги, но от другого автора, мы будем называть прозорливой.

Прозорливость можно измерять как долю рекомендаций, которые далеки от всех оценённых пользователем товаров. Рассмотрим пример с рекомендациями книг. Допустим, мы хотим измерить расстояние $d(b, B)$ между новой книгой b и множеством уже оценённых книг B. Обозначим через $c_{B,w}$ число книг от автора w в множестве B, а через c_B — максимальное количество книг от одного автора в B. Тогда

расстояние можно определить как

$$d(b, B) = \frac{1 + c_B - c_{B,w(b)}}{1 + c_B},$$

где $w(b)$ — автор книги b.

§1.5 Разнообразие

Под разнообразием понимается степень сходства товаров внутри одной пачки рекомендаций (т.е. тех товаров, которые одновременно рекомендуются пользователю). Логично ожидать, что полезность набора из 10 чехлов фотоаппаратов ниже, чем набора из чехла, линзы, объектива, батареек и т.д. — именно это и должна оценивать метрика разнообразия. Можно её задавать как, например, среднее попарное расстояние между товарами в одной пачке. Расстояние может измеряться по каталогу (как далеко в дереве категорий товаров находятся эти два товара) или, например, по аналогии с item-to-item рекомендациями (насколько эти два товара пересекаются по множествам купивших их пользователей).

2 Архитектура рекомендательных систем

В рекомендательной системе может участвовать очень большое количество товаров. При каждом посещении пользователем веб-страницы, где есть блок рекомендаций, необходимо выдать ему k наиболее подходящих товаров, причём достаточно быстро (пользователь не может ждать минуту, пока загрузится страница). В хорошей рекомендательной системе участвуют сотни признаков — их вычисление для каждого товара, а затем ещё и применение ко всем товарам градиентного бустинга или графа вычислений вряд ли получится успеть сделать за 1 секунду. Из-за этого рекомендательные системы работают в несколько этапов: обычно всё начинается с отбора кандидатов, где быстрая модель выбирает небольшое количество (тысячи или десятки тысяч) товаров, а затем только для этих товаров вычисляется полный набор признаков и применяется полноценная модель. В качестве быстрой модели может выступать линейная модель на нескольких самых важных признаках или, например, простая коллаборативная модель.

Список литературы

[1] *Gillis, Nicolas and Glineur, François* (2012). Accelerated Multiplicative Updates and Hierarchical Als Algorithms for Nonnegative Matrix Factorization. // Neural Comput., 24, 4, p. 1085–1105.

[2] *Hu, Yifan and Koren, Yehuda and Volinsky, Chris* (2008). Collaborative Filtering for Implicit Feedback Datasets. // ICDM '08.

[3] *Rendle, S.* (2012). Factorization machines with libFM. // ACM Trans. Intell. Syst. Technol. 3, 3, Article 57.

[4] Field-aware Factorization Machines:
http://www.csie.ntu.edu.tw/~r01922136/slides/ffm.pdf

[5] *Guy Shani, Asela Gunawardana* (2011). Evaluating recommendation systems. // Recommender systems handbook, pp. 257-297. Springer.