

Методы и способы рекомендательных систем

В центре любой рекомендательной системы находится так называемая матрица предпочтений. Это матрица, по одной из осей которой отложены все клиенты сервиса (Users), а по другой – объекты рекомендации (Items). На пересечении некоторых пар (user, item) данная матрица заполнена оценками (Ratings) – это известный нам показатель заинтересованности пользователя в данном товаре, выраженный по заданной шкале (например от 1 до 5).

Summary-based

Это простейшие неперсонализированные системы, в которых потенциальный интерес пользователя определяется просто средним рейтингом товара. По этому принципу работает большинство сервисов, когда пользователь не авторизуется в системе – тот же tripadvisor. Присутствует проблема холодного старта, а также актуальности рекомендаций.

Content-based

Персональные рекомендации предполагают максимальное использование информации о предыдущих покупках/просмотрах/отзывах пользователя. В рамках данного подхода контент сопоставляется с интересами пользователя, полученными из его предыдущих оценок. Чем больше товар/услуга и тп этим интересам соответствует, тем выше оценивается потенциальная заинтересованность пользователя. Очевидное требование здесь — у всех товаров в каталоге должно быть описание.

Исторически предметом Content-based рекомендаций чаще были товары с неструктурированным описанием: фильмы, книги, статьи. Признаками могут быть, например, текстовые описания, рецензии, состав актеров и прочее.

Решается задачей поиска n ближайших соседей и TF-IDF преобразованием.

Collaborative filtering (user based)

Данный класс систем начал активно развиваться в 90-е годы. В рамках подхода рекомендации генерируются на основании интересов других похожих пользователей. Такие рекомендации являются результатом «коллаборации» множества пользователей. Отсюда и название метода.

Классическая реализация алгоритма основана на принципе k ближайших соседей. На пальцах – для каждого пользователя ищем k наиболее похожих на него (в терминах предпочтений) и дополняем информацию о пользователе известными данными по его соседям. Так, например, если известно, что ваши соседи по интересам в восторге от фильма «Кровь и бетон», а вы его по какой-то причине еще не смотрели, это отличный повод предложить вам данный фильм для субботнего просмотра.

Поскольку все пользователи оценивают по-разному – кто-то всем подряд пятерки ставит, а от кого-то четверки редко дождешься – перед расчетом данные лучше нормализовать, т.е. привести к единой шкале, чтобы алгоритм мог корректно сравнивать их между собой.

Collaborative filtering (item based)

Подход Item-based является естественной альтернативой классическому подходу User-based, описанному в первой части, и почти полностью его повторяет, за исключением одного момента — применяется он к транспонированной матрице предпочтений. Т.е. ищет близкие товары, а не пользователей.

Преимущества Item-based перед User-based:

- Когда пользователей много (почти всегда), задача поиска ближайшего соседа становится плохо вычислимой. Item-based подход снижает сложность вычислений.
- Оценка близости товаров гораздо более точная, чем оценка близости пользователей. Это прямое следствие того, что пользователей обычно намного больше, чем товаров и следовательно стандартная ошибка при расчете корреляции товаров там существенно меньше. У нас просто больше информации, чтобы сделать вывод.
- В user-based варианте описания пользователей, как правило, сильно разрежены (товаров много, оценок мало). С одной стороны это помогает оптимизировать расчет — мы перемножаем только те

элементы, где есть пересечение. Но с другой стороны — сколько соседей не бери, список товаров, которые в итоге можно порекомендовать, получается очень небольшим.

- Предпочтения пользователя могут меняться со временем, но описание товаров штука гораздо более устойчивая.

Matrix factorisation

Иногда удобнее описать предпочтения на более «верхнем» обобщенном уровне. Не в формате «он любит фильмы X, Y и Z», а в формате «он любит современные российские комедии». Помимо того, что это увеличит обобщаемость модели, это еще решит проблему большой размерности данных — ведь интересы будут описываться не вектором товаров, а существенно меньшим вектором предпочтений.

Такие подходы называют спектральным разложением или высокочастотной фильтрацией. Одно из наиболее популярных различий матриц называют SVD разложением.

Другие, менее часто используемые подходы.

Ассоциативные правила (Association Rules)

Ассоциативные правила обычно используются при анализе продуктовых корреляций (Market Basket Analysis) и выглядят примерно так «если в чеке клиента есть молоко, то в 80% случаев там будет и хлеб». То есть если мы видим, что молоко в корзину клиент уже положил, самое время напомнить о хлебе.

RBM (restricted Boltzman Machines)

Ограниченные машины Больцмана — относительно старый подход, основанный на стохастических рекуррентных нейронных сетях. Он представляет собой модель с латентными переменными и в этом похож на SVD-разложение. Здесь также ищется наиболее компактное описание пользовательских предпочтений, которое кодируется с помощью латентных переменных. Метод не был разработан для поиска рекомендаций, но он успешно использовался в топовых решениях Netflix Prize и до сих пор применяется в некоторых задачах.

Автоэнкодеры (autoencoders)

В основе лежит все тот же принцип спектрального разложения, поэтому такие сети еще называют denoising auto-encoders. Сеть сначала сворачивает известные ей данные о пользователе в некоторое компактное представление, стараясь оставить только значимую информацию, а затем восстанавливает данные в исходной размерности. В итоге получается некий усредненный, очищенный от шума шаблон, по которому можно оценить интерес к любому продукту.

DSSM (deep semantic similarity models)

Один из новых подходов. Все тот же принцип, но в роли латентных переменных здесь внутренние тензорные описания входных данных (embeddings). Изначально модель создавалась для матчинга запроса с документами (как и content-based рекомендации), но она легко трансформируется в задачу матчинга пользователей и товаров.

Гибридные решения

На практике редко используется только один подход. Для достижения максимального эффекта несколько подходов комбинируются в один.

Два главных преимущества объединения моделей — это увеличение точности и возможность более гибкой настройки на разные группы клиентов. Недостатки — меньшая интерпретируемость и большая сложность реализации и поддержки.

Несколько стратегий объединения:

- Weighting — считать средневзвешенный прогноз по нескольким оценкам,
- Stacking — предсказания отдельных моделей являются входами другого (мета)классификатора, который обучается правильно взвешивать промежуточные оценки,

- Switching — для разных продуктов/пользователей применять различные алгоритмы,
- Mixing — вычисляются рекомендации по разным алгоритмам, а потом просто объединяются в один список.

Например, используется content-based recommender, а в качестве одной из фич — результат коллаборативной фильтрации.