ЕВРАЗИЙСКОЕ НАУЧНОЕ ОБЪЕДИНЕНИЕ
EURASIAN SCIENTIFIC ASSOCIATION
www.esa-conference.ru

Обзор алгоритмов формирования рекомендаций в информационных системах

Ахметов Абай Куатович, магистр естественных наук
ЕНУ им. Л.Н.Гумилева

Аннотация. *В статье описываются алгоритмы формирования рекомендаций в различных информационных системах.*

Ключевые слова. *Коллаборативные системы, корреляция Пирсона, алгоритмы кластеризации, Байесовские сети доверия, цепи Маркова.*

Прежде, чем провести обзор алгоритмов, нужно трезво оценить значимость и понятие термина «рекомендационные системы». Что есть на самом деле рекомендации?

Если зайти на сайт kinopoisk, можно оценивать фильмы по десятибалльной шкале. Оценки суммируются, получается средний рейтинг фильма. На этом же сайте есть блок с рекомендациями для конкретного пользователя. Если пользователь зашел на сайт и оценил несколько фильмов, то система сможет порекомендовать ему еще какие-нибудь фильмы, опираясь на оценки пользователя. Похожая система работает и в социальных сетях. Возьмем, к примеру, популярную в нашей стране сеть Вконтакте, в которой есть раздел «рекомендации по музыке», он может предоставить меломанам рекомендации по их вкусу, основываясь по плейлисту. Тем самым можем предположить, что рекомендации в информационных системах являются очень популярными и используются практически везде: от интернет магазинов до радио приложений.

Формально, рекомендационная система — это программа, которая на основе данных о пользователе (User) и предмете (Item) дает рекомендации. Такая система включает в себя весь процесс — от получения информации до её представления пользователю [1].

На данный момент существует четыре основных типа рекомендационных систем:

- РС, базированные на контенте (Content base)
- Коллаборативные РС (Collaboration)
- РС, базированные на знаниях (Knowlege base)
- Гибридные РС (Gybrid)

Основные шаги в рекомендационной системе, базированной на контенте: проанализировать контент предметов и составить набор его критериев (жанры, тэги, слова), узнать какие критерии нравится пользователю, сопоставить эти данные и получить рекомендации. Критерии объединяют пользователей и предметы в единой системе координат, а тут уже все просто — если точка пользователя и предмета рядом, то вероятно предмет понравится пользователю. Коллаборативные системы в которых рекомендации пользователю рассчитывается на основе оценок других пользователей. Здесь существует множество алгоритмов, но наиболее популярные — User/User (поиск соседей по оценкам), Item/Item (поиск схожести предметов по оценкам пользователей) и SVD (самообучающийся алгоритм).

На рисунке 1 показана упрощенная схема (которая к тому же страдает от разреженности данных по причине использования лишь двух образцов), такое представление весьма удобно.

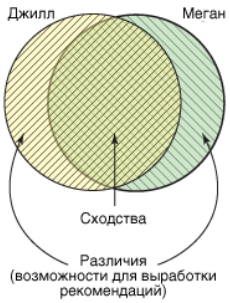

Рис. 1. Сходства и различия, используемые в коллаборативной фильтрации

Системы, базированные на знаниях, в которых для получения рекомендаций используются полученные каким-либо образом знания, чаще всего эти знания добавляются вручную.

Гибридные РС объединяют несколько выше представленных алгоритмов в один.

Прежде чем рассматривать алгоритмы, стоит уделить внимание измерению качества рекомендаций. Так как, если мы хотим улучшить качество рекомендаций, нам нужно научиться его измерять. Для этого алгоритм, обученный на одной выборке — обучающей, проверяется на другой — тестовой. Netflix предложил измерять качество рекомендаций по метрике RMSE:

$$RMSE = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} (\hat{r}_{ui} - r_{ui})^2}$$

Сегодня это стандартная метрика для предсказания оценки. Однако у нее есть свои недостатки:

- У каждого пользователя свое представление о шкале оценок. Пользователи, у которых разброс оценок более широкий, будут больше влиять на значение метрики, чем другие.
- Ошибка в предсказании высокой оценки имеет такой же вес, что и ошибка в предсказании низкой оценки. При этом предсказать оценку 9 вместо настоящей оценки 7 страшнее, чем предсказать 4 вместо 2 (по десятибалльной шкале).
- Можно иметь почти идеальную метрику RMSE, но иметь очень плохое качество ранжирования, и наоборот [2].

Основные используемые алгоритмы для создания РС:

Корреляция Пирсона

Сходство между двумя пользователями (и их атрибутами, такими как статьи, прочитанные в коллекции блогов) может быть точно вычислено с помощью т. н. *корреляции Пирсона*. Этот алгоритм измеряет линейную зависимость между двумя переменными

ЕВРАЗИЙСКОЕ НАУЧНОЕ ОБЪЕДИНЕНИЕ
EURASIAN SCIENTIFIC ASSOCIATION
www.esa-conference.ru

(или пользователями) как функцию их атрибутов. Однако он не вычисляет эту меру по всей совокупности пользователей. Эту совокупность необходимо предварительно отфильтровать до *близких элементов* на основе высокоуровневых показателей сходства, таких как чтение схожих блогов.

Корреляция Пирсона, которая широко применяется в исследовательской деятельности, является весьма популярным алгоритмом в сфере коллаборативной фильтрации.

Алгоритмы кластеризации

Алгоритмы кластеризации— это разновидность т. н. "спонтанного обучения" (unsupervised learning), позволяющая выявить структуру в рядах на первый взгляд случайных (или немаркированных) данных. В общем случае такой алгоритм базируется на выявлении сходства между элементами (например, между читателями блога) посредством вычисления их расстояния от других элементов в пространстве признаков (feature space) (признаком в пространстве признаков может, например, быть количество прочитанных статей в наборе блогов). Количество независимых признаков определяет размерность пространства признаков. Если элементы "близки" друг к другу, то их можно объединить в один кластер.

Существует множество алгоритмов кластеризации. Самым простым из них является алгоритм *k*-средних (*k*-means), который разделяет элементы на *k* кластеров. Первоначально элементы распределяются по этим кластерам в произвольном порядке. Затем для каждого кластера вычисляется *центр масс* (или просто *центр*) как функция его членов. После этого проверяется расстояние каждого члена кластера от центра этого кластера. Если по результатам этой проверки член оказывается ближе к другому кластеру, то он перемещается в этот кластер. После проверки всех расстояний для всех членов центры кластеров вычисляются заново. При достижении стабильного состояния (в процессе очередной итерации члены не перемещались) набор считается кластеризованным надлежащим образом, и алгоритм останавливается.

Вычисление расстояния между двумя объектами может быть трудным для визуализации. Один из распространенных методов решения этой задачи состоит в том, чтобы рассматривать каждый член кластера как многомерный вектор и вычислять для него т.н. евклидово расстояние.

Существует множество других разновидностей кластеризации, в том числе теория адаптивного резонанса (Adaptive Resonance Theory), нечеткая кластеризация методом C-средних (Fuzzy C-means), вероятностная кластеризация с помощью EM-алгоритма (Expectation-Maximization) и т. д.

В рекомендательных механизмах может применяться множество алгоритмов (а если учитывать вариации, то их еще больше). Перечислю некоторые успешно применяемые алгоритмы:

- **Байесовские сети доверия (Bayesian Belief Nets)**— визуально могут быть представлены как ориентированный ациклический граф, ребра которого представляют связанные вероятности переменных.
- **Цепи Маркова (Markov chains)**— основаны на таком же подходе, как у байесовских сетей доверия, но решают проблему выработки рекомендации как последовательную оптимизацию, а не как простое прогнозирование.
- **Классификация по методу Роккио (Rocchio classification)** (основанная на векторной модели) — использует отзывы о релевантности элементов для повышения точности рекомендаций [3].

Возможности сбора данных, которые предоставляет Интернет, существенно упростили использование "мудрости толпы" с помощью коллаборативной фильтрации. С другой стороны, огромное количество доступных данных усложняет реализацию этой возможности. К примеру, поведение некоторых пользователей вполне поддается моделированию, однако другие пользователи не демонстрируют типичного поведения. Наличие таких пользователей может приводить к смещению результатов рекомендательной системы и к снижению ее эффективности. Кроме того, пользователи могут задействовать рекомендательную систему для повышения предпочтительности одного продукта относительно другого продукта — например, посредством отправки позитивных отзывов об одном продукте и негативных отзывов о его конкурентах. Хорошая рекомендательная система обязана справляться с этими проблемами.

Еще одна проблема, свойственная крупным рекомендательным системам, связана с масштабируемостью. Традиционные алгоритмы хорошо работают со сравнительно небольшими объемами данных, однако с ростом этих наборов получение результатов на прежнем уровне качества при помощи традиционных алгоритмов может стать проблематичным. В случае оффлайновой обработки, это может не составлять большой проблемы, однако для сценариев реального времени необходимы более специализированные подходы.

В итоге можно смело сказать, что эта вездесущая система еще требует доработки и создания идеального алгоритма для устранения ошибок и уменьшения погрешностей. Существующие алгоритмы уникальны и должны использоваться каждый для особых случаев. Это означает что универсальных алгоритмов нет, используемые для создания рекомендационных систем в интернет магазинах не подходят для рекомендационных систем сообществ. Но в защиту системы можно сказать, что любая информационная технология со временем стремительно развивается и данная система не исключение.

Литература:

1.Рекомендательные системы: You can (not) advise; https://habr.com/post/176549/

2.Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений: Пер. с англ. - М.: Мир, 1980. - 279 с.

3.Рекомендательные системы. Введение в подходы и алгоритмы: https://www.ibm.com/developerworks/ru/library/os-recommender1/