РАЗРАБОТКА СИСТЕМ РЕКОМЕНДАЦИЙ НА ПРИМЕРЕ ПОСТАВЩИКА ЭЛЕКТРОТЕХНИЧЕСКИХ ТОВАРОВ

Секереш К. В., студент кафедры информационно-аналитических систем СПбГУ,

kost0205@mail.ru

Аннотация

Рекомендательные системы интересны науке и промышленности с появления и распространения web-а. Несмотря на их повсеместное применение в Интернете, востребованными остаются также и оффлайн-версии таких приложений. В данной работе будет рассмотрена конкретная задача рекомендации для покупателей электротоваров.

Введение

Данный документ содержит в себе формализацию задачи создания рекомендательной системы и часть её решения в случае конкретной выборки данных. Рассмотрены основные этапы работы и проблемы, с которыми автор столкнулся, структура предоставленных данных и схема их обработки, а также вопрос формализации заинтересованности покупателя в товаре. Здесь не предполагается никаких низкоуровневых теоретических выкладок; основная цель работы – проверить применимость одного из современных методов создания рекомендательных систем, показать конкретный случай его применения, предложить путь решения практически важной задачи, используя предоставленные выборки.

Читателю, заинтересованному в теоретической стороне вопроса, автор рекомендует ознакомиться с фундаментальными трудами вроде [4], [5] или лекциями на русском языке [6].

Формулировка задачи

Имея историю покупок и классификацию товаров для каждого покупателя, установить, какие товары из ещё не приобретённых будут ему интересны.

Покупателей в дальнейшем будем называть пользователями, согласуясь с терминологией области рекомендательных систем.

Классификация товаров имеет шесть уровней[1] и достаточно подробна:

1 Однако, в большинстве случаев класс товара максимально конкретен на четвёртом или пятом уровнях.

от разбиения на крупные группы (инструмент, светотехника, сопутствующие товары) до полной конкретики (шуруповёрты, гайковёрты, рулетки, …) .

Данные о закупках предоставлены в разрезе по месяцам за период с декабря 2014 года по ноябрь 2015 включительно. Всего в предоставленной выборке присутствует 227,9 тыс. уникальных пользователей и 119 тыс. товаров.

Известны следующие параметры:

- CliCode – код клиента
- RgdCode – код товара
- TerrCode – код территории, откуда отгружен товар
- RgdQuant – количество купленного товара
- QuantCapt – количество обращений за товаром
- RgdClassCode – код класса товара

Идея решения

Рекомендательная система предполагает наличие *оценки* [1] пользователя товаром. Обычно её выражают целым числом из некоторого фиксированного промежутка. Считается, что чем выше оценка пользователя товару, тем более он заинтересован в его покупке. Для пользователя *u* и товара *i* имеем

$$u, i \mapsto r_{ui}$$

Стоит отметить, что данные предметной области не содержат в чистом виде такого отображения и, кроме того, оценка известна (далеко) не для всех пар пользователь/товар.

В связи с этим, работу над решением поставленной задачи можно разбить на три этапа:

- Построение из имеющихся данных (u, i) оценок пользователей r_{ui}
- Предсказание неизвестных оценок $\hat{r}_{ui}$
- По построенным оценкам выбор товара (-ов), рекомендуемых данному пользователю

Далее более подробно рассмотрим выделенные подзадачи.

Построение оценок

Говоря более простым языком, здесь нам нужно оценить заинтересованность пользователей в уже купленных ими товарах, чтобы потом «заглянуть в будущее».

Для составления оценок по имеющимся данным использовалась следующая идея: пользователь тем больше заинтересован в товаре, чем чаще он обращается за ним. Таким образом, будем использовать поле QuantCapt (далее обозначим его за $Q(u,i)$) известных данных. Разумеется, просто просуммировать по этому полю недостаточно, потому что тогда оценки разных пользователей будут несравнимы между собой.

Автор предлагает следующее решение этой проблемы:

$$r_{ui} = \frac{\sum_t Q(u_0, i_0)}{\max_u Q(u, i_0)}$$

Таким образом все оценки для фиксированного товара i_0 буквально приводятся к общему знаменателю. После они домножаются на некоторый коэффициент и округляются[2]. Похожую идею нормализации данных можно увидеть в [2].

Здесь можно отметить, что под i_0 разумно понимать не собственно товар, а некоторый класс товаров, однако на данном этапе автор работает без использования классификации.

В результате применения вышенаписанной формулы, из почти девяти миллионов транзакций получилось 6.7 млн оценок.

Однако, стоит заметить, что применять напрямую к таким оценкам операцию нормировки не всегда корректно. Рассмотрим вырожденный пример, когда пользователь X купил за одно обращение некоторое количество товара, и больше за ним не обращался. Тогда числитель в формуле (1) будет равен единице, и, если положить знаменатель достаточно большим, в качестве «оценки» нам предстанет некоторое близкое к нулю число, которое после округления даст единичную оценку, такую же, как если бы X совершил 20% обращений по сравнению с максимумом!

2 В формуле этого не показано.

Добавив сюда наблюдение о большом размере выборки, автор сократил её, оставив лишь товары, удовлетворяющие следующим критериям:

1. Размах[3] оценок за данный товар должен быть не менее 5.
2. За данный товар в выборке присутствует не менее 50 оценок от разных пользователей.

Итоговый размер тестовой выборки – 5.8 млн оценок (218.2 тыс. пользователей и всего 10 тыс товаров)[4]. Нужно заметить, что такое малое количество оставшихся товаров вполне согласуется с поставленной задачей: от рекомендательной системы ожидается товар, популярный у пользователей.

Предсказание оценок

Для решения этой подзадачи разработано множество алгоритмов, в реализации которых в рамках данного документа мы вдаваться не будем. Автор планирует использовать так называемую *коллаборативную фильтрацию* [1], [2], то есть предсказывающие алгоритмы, основанные на действиях похожих пользователей. Например, при покупке компьютера очень часто (почти всегда) клиент приобретает клавиатуру и мышь. Основываясь на этой и других похожих аналогиях, система будет предсказывать высокую оценку для мыши тем пользователям, что купили (поставили высокую оценку) компьютер, но ещё не купили мышь.

Особо остановиться нужно, пожалуй, на том, как оценивается качество таких предсказаний. Для этого требуется немного «обмануть» систему, и потребовать от неё предсказание оценок, которые нам *уже известны*. Такие оценки назовём тестовыми данными и обозначим через $\hat{R}$. Далее мы сможем некоторым образом оценить ошибку. Рассмотрим здесь следующие способы:

- Средний модуль ошибки:

$$\mathrm{MAE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

3 Размахом называют разницу между наибольшим и наименьшим значением.

4 Возможно, в дальнейшем выборка будет ещё более сокращена, если результат последнего этапа деятельности окажется неудовлетворительным.

- Среднеквадратичное отклонение:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}.$$

  Очевидно, чем меньше значения оцениваемой метрики, тем лучше.

Чтобы более уверенно интерпретировать результаты (то есть получившиеся значения метрик ошибок), как правило, создают даже несколько множеств тестовых данных[5], одинаковых по мощности, а потом сравнивают.

Именно так мы и поступим в следующем разделе.

Получение решения

Здесь идея довольно очевидна — те оценки, которые мы получили на предыдущем шаге, считаются настоящими. Далее просто выбирается товар с максимальной оценкой. При этом также можно учитывать информацию о классификации: например, не рекомендовать более одного товара из класса.

Полученные результаты

На данный момент реализовано получение и предсказание оценок (первые два этапа), значения ошибки получились следующими (на трёх различных порциях тестовых данных):

	Порция 1	**Порция 2**	**Порция 3**
RMSE	0.084	0.084	0.084
MAE	0.254	0.254	0.254

Таблица 1: Показатели метрик ошибки, полученные на одном из тестовых запусков

Размер одной порции данных составлял примерно **1.45 млн** оценок – четверть от общего размера выборки.

Отметим, что на трёх разных порциях данных система работает стабильно, давая одну и ту же ошибку. Кроме того, средняя ошибка составляет всего 0.2 балла из пяти, это гораздо меньше, чем ожидалось.

5 Эти множества (folds) могут быть как фиксированными, так и случайными от запуска к запуску.

Стоит обратить внимание, что низкое значение средней ошибки, вообще говоря, может и не означать хорошую работу системы в целом, нужно проверять, какие конкретно товары она будет предсказывать.

Заключение

Описываемый подход показал себя, по крайней мере, достойным применения и анализа. В дальнейшем будет получена демонстрация работы всей системы на некотором множестве пользователей, и сравнение результата предсказаний с их прошлыми покупками. Последнее можно делать как вручную, так и автоматически, используя предоставленную классификацию.

Список литературы

[1] Basics of user based collaborative filter in predictive analysys, A. Bari, M. Chaouchi, T. Jung

[2] A Preprocessing Method for Improving Effectiveness of Collaborative Filtering G. Kim, J. Chun, Ph.D. Sang-goo Lee, Ph.D.

[3] Mining of massive datasets, Jeffrey D. Ullman, Chapter 9.

[4] Recommender Systems Handbook. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., Chapter 5.

[5] Item-Based Collaborative Filtering Recommendation Algoritm, B. Sarwar, G. Karypis, J. Konstan, J. Riedl

[6] https://habrahabr.ru/company/yandex/blog/241455/ – лекция о рекомендательных системах в Яндексе.

Контакты

Связаться с автором можно следующими способами:

- Электронная почта: kost0205@mail.ru
- Telegram – начать переписку можно по ссылке: t.me/faerics