

КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ SVD++

Введение. С развитием электронной торговли возникла необходимость в предоставлении рекомендаций, полученных в результате фильтраций всего спектра доступных альтернатив. Пользователям трудно выбрать наиболее подходящие варианты из огромного разнообразия товаров и услуг, которые предоставляют веб-сайты и веб-сервисы. Рекомендательные системы представляют собой программные инструменты и методы, обеспечивающие предложения по предметам, которые, скорее всего, представляют интерес для конкретного пользователя. Предложения относятся к различным процессам принятия решений: какие предметы купить, какую музыку послушать, какие новости читать. В своей простейшей форме персональные рекомендации предлагаются в качестве списка предметов и их рейтингов. При вычислении этого рейтинга пытаются определить наиболее подходящие продукты или услуги в зависимости от предпочтений и ограничений пользователя. Для выполнения такой вычислительной задачи собирают информацию о предпочтениях пользователей, которые либо выражены явно, например, в качестве оценки для продуктов, или выводятся путем интерпретации действий пользователя. Например, рекомендательная система может рассмотреть факт просмотра конкретной страницы веб-сайта как неявный признак предпочтения элементов, показанных на этой странице.

Основная часть. Одной из наиболее популярных техник для построения персонализированных рекомендательных товарных систем является коллаборативная фильтрация (Collaborative Filtering). Коллаборативная фильтрация вырабатывает рекомендации, основанные на модели предшествующего поведения пользователя. Эта модель может быть построена исключительно на основе поведения данного пользователя или (что более эффективно) с учетом поведения других пользователей со сходными характеристиками. В тех случаях, когда коллаборативная фильтрация принимает во внимание поведение других пользователей, она использует знание о группе (group knowledge) для выработки рекомендаций на основе подобия пользователей. По существу, рекомендации базируются на автоматическом сотрудничестве множества пользователей и на выделении (методом фильтрации) тех пользователей, которые демонстрируют схожие предпочтения или шаблоны поведения.

В качестве примера предположим, что системные администраторы сталкиваются с проблемой рекомендации товаров пользователям на основе информации от многих других пользователей, которые подписываются на блоги и просматривают их. Например, можно объединить в одну группу пользователей, которые читают несколько одних и тех же блогов. По этой информации идентифицируются самые популярные блоги среди тех, которые читают участники этой группы. Затем конкретному пользователю этой группы администраторы рекомендуют самый популярный блог из тех, на который он еще не подписан.

В таблице (рисунок 1) строки соответствуют набору блогов, а столбцы — пользователям. Ячейка на пересечении строки блога и столбца пользователя содержит количество статей, прочитанных этим пользователем в этом блоге. Кластеризация пользователей на основе читательских привычек позволяет выделить два кластера, каждый из которых содержит по два пользователя. Схожесть читательских привычек у членов каждого кластера: кластер 1 образуют участники с именами Марк и Элиза, каждый из которых прочел по несколько статей по тематике “Linux” и по тематике «Облачные вычисления»; кластер 2 образуют участники с именами Меган и Джилл, каждый из которых прочел по несколько статей по тематике “Java-технологии” и по тематике “Agile-технологии”.

Теперь системные администраторы могут идентифицировать определенные различия в рамках каждого кластера и сформировать значимые рекомендации. В кластере 1 Марк прочел 10 статей из блога по продуктам с открытым исходным кодом, а Элиза не прочла ни одной такой статьи; Элиза прочла одну статью в блоге по Agile-технологиям, а Марк не прочел ни одной такой статьи. Таким образом, Элизе можно порекомендовать блог по продуктам с открытым исходным кодом. Для Марка невозможно сформировать никаких рекомендаций, поскольку небольшая разница между ним и Элизой по количеству прочтений блога по Agile-технологиям с большой вероятностью будет отфильтрована. В кластере 2 Джилл прочла три статьи в блоге по продуктам с открытым исходным кодом, а Меган не прочла ни одной такой статьи; Меган прочла три статьи в блоге по Linux, а Джилл не прочла ни одной такой статьи. Таким образом, для участников кластера 2 можно сформировать следующие две рекомендации: участнику Джилл рекомендуется блог по Linux, а участнику Меган — блог по продуктам с открытым исходным кодом.

Блоги	Марк	Меган	Элиза	Джилл
Linux	13	3	11	-
Продукты с открытым кодом	10	-	-	3
Облачные вычисления	6	1	9	-
Java-технологии	-	6	-	9
Agile-технологии	-	7	1	8
Количество статей, прочитанных данным пользователем				
Кластер	1	2	1	2

Рисунок 1 — Простой пример коллаборативной фильтрации

Другой способ рассмотрения этих отношений основан на их сходствах и различиях, как иллюстрирует диаграмма Венна (рисунок 2). Сходства определяют, по каким признакам следует группировать пользователей, имеющих схожие интересы. Различия — это возможности, которые могут быть использованы для выработки рекомендаций, например, посредством применения фильтра популярности.

Коллаборативную фильтрацию можно разделить на два основных подхода: подход, основанный на соседстве и подход, основанный на модели. Подход, основанный на соседстве, является исторически первым в коллаборативной фильтрации и используется во многих рекомендательных системах. В данном подходе для активного пользователя подбирается подгруппа пользователей, схожих с ним. Комбинация весов и оценок подгруппы используется для прогноза оценок активного пользователя. Подход, основанный на модели, предоставляет рекомендации, измеряя параметры статистических моделей для оценок пользователей, построенных с помощью таких методов, как метод байесовских сетей, кластеризации, латентной семантической модели, такие как сингулярное разложение, вероятностный латентный семантический анализ и др. Модели разрабатываются с использованием интеллектуального анализа данных, алгоритмов машинного обучения, чтобы найти закономерности на основе обучающих данных. Число параметров в модели может быть уменьшено в зависимости от типа с помощью метода главных компонент.

Для того чтобы получить рекомендации, системы коллаборативной фильтрации товаров увязывают две основных сущности: пользователей и товары. Простейший способ связи — это явно указанный пользователем рейтинг (оценка) товара. Если есть данные об оценке товаров различными пользователями, то получаем матрицу «пользователь—товар». В первую очередь необходимо определить базовые предикторы (baseline predictors) $b_{i,a}$, которые складываются из базовых предикторов отдельных пользователей b_i и базовых предикторов отдельных продуктов b_a , а также просто общего среднего рейтинга по базе μ :

$$b_{i,a} = \mu + b_i + b_a.$$

Если нужно найти только базовые предикторы, необходимо будет найти такие μ , b_i и b_a , для которых $b_{i,a}$ лучше всего приближают имеющиеся рейтинги. Затем нужно будет добавить факторы; теперь, когда сделали поправку на базовые предикторы, остатки будут сравнимы между собой, и можно будет получить разумные факторы, формула 2:

$$\hat{r}_{i,a} = \mu + b_i + b_a + v_a u_i,$$

где v_a — вектор факторов, представляющий продукт a , а u_i — вектор факторов, представляющий пользователя i .

Для того чтобы определить наименьшую погрешность на имеющихся данных, необходимо использовать сумму квадратов отклонений тех рейтингов, которые уже известны:

$$L(\mu, b_i, b_a, u_i, v_a) = \sum_{(i,a) \in D} (r_{i,a} - \hat{r}_{i,a})^2 = \sum_{(i,a) \in D} (r_{i,a} - \mu - b_i - b_a - v_a u_i)^2.$$

Для того чтобы минимизировать функцию погрешности, необходимо пользоваться градиентным спуском: берём частные производные по каждому аргументу и двигаемся в сторону, обратную направлению этих частных производных. Функция ошибки — квадратичная поверхность. В конечном итоге получаем функцию ошибки:

$$b_*, q_*, p_* = \arg \min_{b,q,p} \sum_{(i,a)} (r_{i,a} - \mu - b_i - b_a - q_a p_i)^2 + \gamma (\sum_i b_i^2 + \sum_a b_a^2 + \|q_a\|^2 + \|p_i\|^2),$$

где γ — параметр регуляризации.

Если взять частные производные по каждой из оптимизируемых переменных, получим простые правила для градиентного спуска:

$$\begin{aligned} b_i &:= b_i + \tau(e_{i,a} - \gamma b_i), \\ b_a &:= b_a + \tau(e_{i,a} - \gamma b_a), \\ q_{a,j} &:= q_{a,j} + \tau(e_{i,a} p_{i,j} - \gamma q_{i,j}), \\ p_{i,j} &:= p_{i,j} + \tau(e_{i,a} q_{a,j} - \gamma p_{i,j}) \end{aligned}$$

для всех j , где $e_{i,a} = r_{i,a} - \hat{r}_{i,a}$ — ошибка на данном тестовом примере, а τ — скорость обучения.

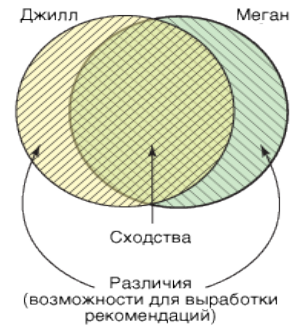


Рисунок 2 — Сходства и различия, используемые в коллаборативной фильтрации

Описанная выше модель называется SVD. Модель SVD++ основывается на модели SVD. Точность предсказания рейтингов увеличивается за счёт добавления в модель неявного отклика, который представляет дополнительные индикаторы предпочтений пользователя. Это весьма полезно для тех пользователей, которые предоставили больше неявных отзывов, чем явных.

Заключение. Большинство крупномасштабных коммерческих и социальных веб-сайтов рекомендуют своим пользователям различные товары и услуги, например, товары для дальнейшего изучения или людей, к которым целесообразно обратиться. Рекомендательные механизмы сортируют огромные объемы данных для выявления потенциальных предпочтений пользователей.

Важно понимать, что существуют другие подходы к коллаборативной фильтрации, другие формулы для вычисления оценок продуктов, другие меры схожести. Дальнейшее развитие этой идеи можно вести в следующих направлениях:

1) оптимизация используемых структур данных. В практических задачах для хранения данных можно использовать разреженные матрицы, а вместо текстовых имен пользователей и названий продуктов использовать числовые идентификаторы (пронумеровать всех пользователей и все продукты);

2) оптимизация производительности. Вычислять рекомендацию при каждом обращении пользователя — занятие крайне затратное. Есть несколько вариантов обхода этой проблемы: кластеризация пользователей и вычисление меры схожести только между пользователями, принадлежащими одному кластеру, и вычисление коэффициентов схожести «продукт—продукт». Для этого нужно транспонировать матрицу «пользователь—продукт», после чего для каждого из продуктов вычислить набор наиболее схожих продуктов;

3) подбор меры сходства. Косинусная мера является одной из часто используемых, но выбор меры нужно производить только по результатам анализа данных системы;

4) модификация алгоритма фильтрации. Возможно, другой алгоритм фильтрации даст более точные рекомендации в конкретной системе. Опять же, сравнение различных алгоритмов можно производить только в применении к конкретной системе.

УДК 004:001

М. Ю. Колядко

Учреждение образования «Барановичский государственный университет», Барановичи

ВЗГЛЯД СТУДЕНТА НА ПРИОРИТЕТЫ В ИЗУЧЕНИИ И ИСПОЛЬЗОВАНИИ КОМПЬЮТЕРНЫХ ПРОГРАММ НА МЛАДШИХ И СТАРШИХ КУРСАХ

Введение. Ознакомление в сети Интернет с набором программных средств, изучаемых на инженерных факультетах университетов Беларуси, России, Украины, выявляет их разнообразие: ANSYS, ABAQUS, NASTRAN, EMAS, ADAMS, AutoCAD, UNIGRAPHICS NX, КОМПАС, T-FLEX, CADMECH, Solid Works, AutoCAD Inventor, Autodesk Mechanical Desktop, Solid Edge, КОМПАС-3D, CATIA, ProEngineer, MathCAD, MATLAB и др.

Основная часть. Часть из перечисленных программ предназначена для моделирования полей физических величин, полей с использованием метода конечного элемента (ANSYS, ABAQUS, NASTRAN, NX NASTRAN, LS-DYNA, ADAMS, T-FLEX и др.) и требует от пользователя серьезных знаний по общенаучным и общетехническим дисциплинам. Для специализированных программ необходимы знания по узкопрофессиональным дисциплинам. Так, при разработке и эксплуатации систем с программным управлением (роботы, специализированные станки) приходится использовать программы SprutCAM, ADEM, EdgeCAM, PowerMill, Mastercam и др.

Весьма многообразно существующее программное обеспечение для автоматизированного конструирования для машиностроения принято классифицировать по функциональной полноте. По этому признаку оно делится условно на три уровня. К нижнему уровню (легкие САПР) относятся программы для автоматизации разработки и сопровождения технической документации, реализующие 2D-модели в виде чертежей и спецификаций, технологических карт, ведомостей. Например: AutoCAD LT (AutoDesk), T-Flex CAD 2D (Топ Системы), КОМПАС-График (Аскон), CADMECH (Интермех) и др. На среднем (средние САПР) уровне располагаются программные комплексы, которые позволяют создать трехмерные параметрические модели сравнительно несложного изделия методом твердотельного моделирования, выполнять проверочные расчеты деталей и сборок. К числу этих программных комплексов можно отнести: Solid Works (SolidWorks Corp., США), AutoCAD Inventor (AutoDesk, США), Autodesk Mechanical Desktop (AutoDesk, США), Solid Edge (Siemens PLM Software, Германия), T-FLEX CAD/CAM/CAE/CAPP/PDM (Топ Системы, Россия), КОМПАС-3D (Аскон, Россия) и др. Эти САПР относятся к категории индивидуального пользования.