

МЕХАНИЗМ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шнейдер В. В.

Теслюк В. Н. – канд. физ.-мат. наук, доцент

Механизм рекомендаций (иногда называемый рекомендательной системой) - это инструмент, который позволяет разработчикам алгоритмов прогнозировать, что пользователю может понравиться или не понравиться в списке заданных элементов.

Механизм рекомендаций работает одним из двух способов. Он может полагаться на свойства элементов, которые нравятся пользователю, и анализирует их, чтобы определить, что еще можно ему предложить; или он может полагаться на симпатии и антипатии других пользователей, которые затем использует для вычисления индекса подобия между пользователями и соответственно рекомендовать им элементы. Также возможно объединить оба этих метода для создания более надежного механизма. Однако, как и во всех других проблемах связанных с информацией, важно выбрать алгоритм, подходящий для решения проблемы.

Для работы механизма понадобятся два типа данных – идентификатор пользователя и элемент, который необходимо анализировать. Поэтому нет необходимости принимать во внимание другие атрибуты элемента при выработке рекомендаций. Сходство между двумя пользователями представлено индексом подобия - десятичным числом от -1,0 до 1,0. А соотношение между пользователем и элементом будет представлено с использованием другого десятичного числа от -1,0 до 1,0 (другими словами это выражение отношения пользователя к элементу – не нравится или нравится).

Данный механизм состоит из нескольких наборов. У каждого пользователя будет два набора: элементы, которые понравятся пользователю, и те элементы, которые не понравятся. У каждого элемента также есть два набора, связанных с ним: пользователи, которым понравился элемент, и пользователи, которым он не понравился.

Чтобы вычислить индекс подобия, мы будем использовать вариацию формулы индекса Жаккара. Первоначально она известна как «коэффициент сходства» (придуман Полом Жаккарром), формула сравнивает два набора и дает простую десятичную статистику между 0 и 1,0:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Формула включает деление числа общих элементов в любом наборе на число всех элементов (учитываемых только один раз) в обоих наборах. Индекс Жаккара двух одинаковых множеств всегда будет равен 1, а индекс из двух множеств без общих элементов всегда даст 0.

Как уже говорилось ранее, с точки зрения системы пользователями являются три сущности: идентификатор, набор элементов, которые понравились и набор элементов, которые не понравились. Если необходимо определить индекс сходства пользователей, основанный только на наборе понравившихся элементов, то можно было бы непосредственно использовать формулу индекса Жаккара:

$$S(U_1, U_2) = \frac{|L_1 \cap L_2|}{|L_1 \cup L_2|}$$

Здесь U_1 и U_2 - это два пользователя, которых мы сравниваем, а L_1 и L_2 - наборы элементов, которые понравились U_1 и U_2 , соответственно. Следовательно, два пользователя, не одобряющие одни и те же элементы, также должны быть похожи. Немного модифицируем уравнение:

$$S(U_1, U_2) = \frac{|L_1 \cap L_2| + |D_1 \cap D_2|}{|L_1 \cup L_2 \cup D_1 \cup D_2|}$$

Вместо того, чтобы учитывать общие симпатии в числителе формулы, добавим количество общих антипатий – D_1 и D_2 . В знаменателе возьмем количество всех предметов, которые понравились или не понравились пользователю. Также необходимо учесть случай, когда два пользователя являются противоположностями в своих предпочтениях. Индекс подобия двух пользователей, которым нравится один элемент, а другой - не нравится, не должен быть 0:

$$S(U_1, U_2) = \frac{(|L_1 \cap L_2| + |D_1 \cap D_2| - |L_1 \cap D_2| - |L_2 \cap D_1|)}{|L_1 \cup L_2 \cup D_1 \cup D_2|}$$

В числителе вычитается количество конфликтующих симпатий и неприязнь двух пользователей из числа их общих симпатий и антипатий. Это приводит к тому, что формула индекса подобия имеет диапазон значений от -1,0 до 1,0. У двух пользователей, имеющих одинаковые вкусы, индекс сходства будет 1,0, тогда как у двух пользователей, имеющих полностью противоречивые вкусы в фильмах, будет показатель сходства -1,0.

После того, как будут получены индексы подобия пользователей, можно найти вероятность того, что пользователю U понравится элемент M:

$$P(U, M) = \frac{Z_L - Z_D}{|M_L| + |M_D|}$$

Z_L и Z_D представляют собой сумму индексов подобия пользователя U со всеми пользователями, которым понравился или не понравился элемент M. $|M_L| + |M_D|$ - общее количество пользователей, которым понравился или не понравился элемент M. Результат вычисления $P(U, M)$ дает число от -1,0 до 1,0.

Данная формула является механизмом рекомендаций. Алгоритм прост и понятен, а также его можно улучшать и расширять под собственную систему. На данной фундаментальной идее строятся многие рекомендательные библиотеки, позволяющие работать не только с маленькими выборками, но и большими базами данных, что делает данный механизм очень популярным в современных быстрорастущих системах.

Список использованных источников:

1. Розенберг Г. С. Поль Жаккар и сходство экологических объектов, М.: ИзВБРАН, 2012, 202 с.
2. Джанак Д., Занкер М., Фелферниг А., Фридрих Г. Введение в рекомендательные системы, М.: Издательство Кембриджского университета, 2010, 775 с.

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЕСКТОП ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ TITANIUM

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Цегельник Н.Г.

Поттосина С. А. – канд. физ.-мат. наук, доцент

Сфера информационных технологий приобрела особое значение в современном мире. Быстрое развитие систем разработки программного обеспечения и сетевых технологий привело к увеличению производства на рынке программного обеспечения. Усиление конкуренции между производителями программного обеспечения требует повышенного внимания к качеству продукции. Большое количество компаний по всему миру начали инвестировать в повышение качества программного обеспечения. Тестирование позволяет определить, соответствует ли программный продукт предъявляемым к нему требованиям, выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации. Своевременное выявление и исправление ошибок и недоработок имеет огромное значение в процессе разработки программного продукта, поскольку это уменьшает риски и при этом происходит снижение затрат на разработку программного обеспечения.

Анализируя рациональность введения автоматизированного тестирования в рамки проекта, необходимо оценить: степень сложности выполнения тестовых сценариев вручную, частоту запуска написанных тестов, необходимость в увеличении скорости прохождения тестов. Если все же решено автоматизировать тестовые сценарии, то необходимо ли это выполнять для всего набора тестов, или достаточно сделать частичную автоматизацию. Внедрение автоматизированного тестирования позволяет решить такие проблемы, как снижение времени, необходимого для выполнения тестирования, а также повышение качества процесса тестирования.

Автоматизация тестирования позволяет ускорить процесс тестирования, в особенности такие аспекты:

- часто используемую функциональность, проверку критической функциональности, где риски возникновения ошибок достаточно высоки;
- рутинные операции (проверка форм, в которых необходимо заполнять огромное количество полей);
- сложно доступные места в системе, такие как запись данных в базу, бекэнд процессы;
- проверка корректного поиска данных;