УДК 004.85, 004.89

МЕТОД СКОЛЬЗЯЩЕГО КОНТРОЛЯ ДЛЯ ОЦЕНКИ КАЧЕСТВА РЕКОМЕНДАТЕЛЬНЫХ ИНТЕРНЕТ-СЕРВИСОВ

Д.И. Игнатов (*dignatov@hse.ru*)

А.Ю. Каминская (*skam90@gmail.com*)

Р.А. Магизов (*dzhamin@gmail.com*)

ГУ-ВШЭ, Москва, Покровский бульвар д.11, 109028

В статье описываются современные алгоритмы, применяемые в рекомендательных системах. Предлагается способ оценки их качества в терминах точности и полноты предсказаний. Данная методика основывается на технике скользящего контроля. Проведены эксперименты на реальных и синтетических наборах данных. Результаты позволяют выявить лучший метод среди исследуемых и подобрать его параметры.

Введение

Современному Интернет-пользователю довольно часто приходится сталкиваться с рекомендательными системами. Например, приобретая книгу *X* в Интернет-магазине, пользователь такой системы получает рекомендации вида «другие пользователи вместе с книгой *X* приобрели также книги *Y* и *Z*». Также существует множество сервисов для рекомендации пользователю интересных сайтов, известных как сервисы социальных закладок. Нетехнологичным способом получения рекомендаций можно назвать совет друзей и знакомых. С ростом числа вещей, для выбора которых вам нужна рекомендация, мнения знакомых становится недостаточно, они могут просто не знать о новинках. В этой затруднительной ситуации помогает так называемая коллаборативная фильтрация [Golberg et al., 1992]. Методы рекомендательных систем, основанные на коллаборативной фильтрации, часто работают по довольно простой схеме. Они просматривают большую группу людей, находят тех из них, предпочтения которых близки к вашим, а затем создают список выбранных ими вещей и ранжируют его, в итоге предлагая вам *N* верхних предметов такого списка. Другим менее очевидным, но востребованным

приложением является рекомендация словосочетаний в системах контекстной рекламы, где фирмы приобретают рекламные словосочетания, по которым поисковые системы могут находить их и выдавать рекламу на такой запрос.

1. Существующие модели рекомендательных систем

В нашей работе представлены две группы методов, ставших классическими в рассматриваемой области: user-based (основанные на сходстве пользователей) и item-based (основанные на сходстве признаков) (см. [Badrul et al., 2000]). Ключевым для работы этих методов является понятие сходства (математически это может быть мера Жаккара, корреляция Пирсона, мера косинусов и т.п.). Исходные данные представлены объектно-признаковыми матрицами, в которых столбцам соответствуют признаки (товары), а строкам объекты (пользователи). На пересечении строк и столбцов записаны 1 или 0, которые означают факт покупки или ее отсутствие. Также значениями ячеек таблицы могут быть рейтинги или оценки товаров, например, фильмов, поставленные пользователями.

1.1. Рекомендации, основанные на сходстве пользователей (User-based)

В методах группы user-based сходство ищется между пользователями и в качестве рекомендаций пользователю u_0 выдается n самых часто покупаемых товаров k наиболее похожими на него покупателями. Введем обозначения: u_0 – «целевой» пользователь, u_0^I – предметы, которые он оценивал, sim(u_0, u)– сходство пользователя u_0 с пользователем u. В нашей работе для подсчета сходства мы использовали коэффициент корреляции Пирсона. Определим множество ближайших соседей (соседство) для целевого пользователя по формуле:

$$N(u_0) = \{u \mid sim(u_0, u) \geq \Theta\}.$$

На самом деле удобнее брать top-k ближайших соседей, т.е. top-k определяет Θ. Таким образом, во множество ближайших соседей попадают те пользователи, сходство которых с целевым выше некоторого порога. После упорядочения пользователей по убыванию сходства, нужно отбирать не только первые k, но и проверить значение сходства у следующего *(k+1)* по списку. Если это значение совпадает с последним, включать и $k+1$-го пользователя в соседство. И так до тех пор, пока сходство не изменится. Поскольку мы предсказываем оценку целевым пользователем конкретного предмета i, нам интересны только те пользователи из этого соседства, которые оценивали предмет i:

$$N(u_0 \mid i) = \{u \mid i \in u^I \ \& \ u \in N(u_0)\}.$$

Обозначив оценку предмета i пользователем u через r_{ui}, получаем итоговую формулу для предсказанной оценки:

$$\hat{r}_{u_0 i} = \frac{\sum_{u \in N(u_0|i)} sim(u_0, u) \times r_{ui}}{\sum_{u \in N(u_0|i)} sim(u_0, u)} .$$

Чтобы выдать пользователю рекомендацию, нужно посчитать такие предсказания для каждого из неоцененных им предметов, ранжировать по убыванию и выдать первые n предметов в качестве рекомендованных.

1.2. Рекомендации, основанные на сходстве признаков (Item-based)

Идея метода item-based аналогична идее вышеописанного метода user-based, однако сходство уже ищется по признакам, то есть по предметам. Введем обозначения: u_0 – «целевой» пользователь, u_0^I – предметы, которые он оценивал, $sim(i,j)$ – сходство товара i с товаром j. Определим соседство для предмета i аналогично тому, как определяли его для целевого пользователя: $N(i) = \{ j \mid sim(i, j) \geq \Theta \}$.

Это top-k ближайших к i товаров, top-k определяет Θ. Чтобы предсказать оценку целевому пользователю, нужно сравнивать те предметы, которые он оценивал, с теми, которые нет. Поэтому «уточним» формулу для соседства в отношении к целевому пользователю:

$$N(i \mid u_0) = \{ j \mid j \notin u_0^I, i \in u_0^I, j \in N(i) \} .$$

Обозначим r_{ui} оценку предмета i пользователем u, таким образом, итоговая формула имеет вид:

$$\hat{r}_{u_0 i} = \frac{\sum_{j \in u_0^I} sim(i, j) \times r_{u_0 j}}{\sum_{j \in u_0^I} sim(i, j)} .$$

Далее ранжируем оценки по убыванию и выдаем первые n в качестве рекомендаций.

Преимущества этому методу дает тот факт, что обычно на сайтах онлайновой торговли количество пользователей постоянно увеличивается, в то время как новые предметы добавляются не так часто. Поэтому вычисление сходства пользователей каждый раз при формировании рекомендации может занимать много времени, а сходство предметов

можно посчитать заранее в режиме офлайн, и потом использовать эту матрицу.

У этих методов есть ряд недостатков, например, в случае «холодного старта», когда нам ничего неизвестно о новом клиенте, такие системы не смогут ничего порекомендовать, но они успешно справляются со своей задачей, когда известна история пользователя.

2. Меры сходства

Чтобы определить сходство между объектами или признаками, используют различные *меры (метрики) сходства.* Обычно эти величины принимают значения от 0 до 1 (при абсолютном сходстве). Рассмотрим некоторые из таких метрик.

2.1. Сходство, основанное на расстоянии.

Для определения сходства можно сначала вычислить расстояние одним из способов:

2.1.1. Евклидово расстояние.

В данном случае каждый объект представляется как вектор в пространстве признаков (или наоборот). Тогда между любыми двумя точками можно найти Евклидово расстояние по известной формуле.

$$d = \sqrt{\sum_i \left(x_i - y_i\right)^2}$$

2.1.2. Расстояние Хемминга.

Эта метрика применяется для бинарных данных. Расстояние в данном случае равно количеству разных значений координат двух векторов.

$$d = \sum_{x_i \neq y_i} 1$$

Для выбранного расстояния сходство часто ищется по формуле:

$$s = \frac{1}{1+d}.$$

$$s = \frac{1}{1+d}.$$

Вернемся к метрике Хемминга. Поскольку d может принимать только натуральные значения и 0, максимальное значение s равняется 1 (при нулевом расстоянии), но уже при $d = 1$ значение $s = 1/2$, что является очевидным недостатком: если имеется два 100-мерных бинарных вектора, у которых различие лишь в одной из координат, то согласно формуле получаем, что они сходны лишь наполовину. Такой «несглаженный» характер значений сходства можно видеть на графике:

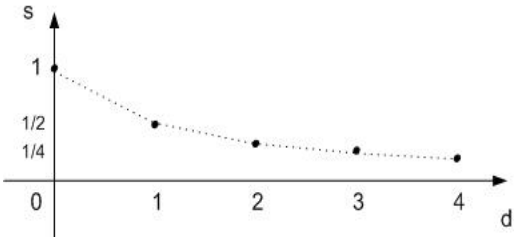

Рис.1. Зависимость сходства от расстояния Хемминга

2.2. Корреляции как сходства

В данном случае сходство считается с помощью известной формулы для корреляции из математической статистики (корреляция Пирсона):

$$Pearson = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \cdot \sum_i (y_i - \overline{y})^2}}.$$

Главным недостатком этого способа определения сходства является то, что корреляция Пирсона не определена для векторов с постоянными значениями. Действительно, если мы имеем вектор $x = (4, 4, ..., 4)$, знаменатель становится равным нулю.

Именно поэтому мы можем терять рекомендации. В качестве примера рассмотрим два вектора: $a = (0, 5, 5, 4)$ и $b = (0, 4, 5, 0)$. Если представить эти векторы как набор оценок двух пользователей, интуитивно понятно, что такие пользователи достаточно схожи. Однако корреляция посчитана не будет, поскольку она считается только по тем координатам, которые не равны нулю у обоих векторов. В нашем примере нужно посчитать корреляцию для $(5, 5)$ и $(4, 5)$. Но, как было показано выше, для первого вектора она не определена. Некоторые авторы предлагают в данном случае считать корреляцию нулевой [Сегаран, 2008], но на наш взгляд, это некорректно.

2.3. Другие меры сходства

Для определения сходства существует несколько десятков разных мер. Например, *косинусная мера или коэффициент Жаккара.*

3. Методика оценивания качества рекомендаций

В этой работе мы предлагаем методику оценки качества рекомендаций, выдаваемых такими системами. Пусть исходные данные представлены объектно-признаковой таблицей, бинарное отношение $T \subseteq U \times I$ показывает,

совершил ли пользователь $u \in U$ покупку $i \in I$, что выражается как uTi. Для оценки качества рекомендаций в терминах точности и полноты можно применить подход, использующий разбиение исходной выборки (множество U) на два подмножества: тестовое U_{test} и обучающее $U_{training}$. Размер тестового множества, как правило, меньше размера обучающего, например, 20% и 80% соответственно. Полнота и точность поиска оценивается на тестовой выборке. Каждый элемент u тестовой выборки U_{test} в свою очередь разбивается на две части, на оцененные признаки $I_{visible}$ и неоцененные I_{hidden}, т.е. такие, которые мы намеренно скрываем. Отметим, что в существующей научной литературе отношение размеров $I_{visible}$ и I_{hidden} не обсуждается. Далее, например, алгоритм рекомендаций по пользователям использует сходство объектов из тестовой и обучающей выборки для формирования рекомендаций. Каждый пользователь из U_{test} получает рекомендации в виде множества фиксированного размера $r_{\text{n}}(u)=\{i_1,i_2,\ldots,i_{\text{n}}\}$. Далее точность и полнота вычисляются следующим образом:

$$recall = \frac{|\, r_n(u) \cap u^I \cap I_{hidden} \,|}{\left| u^I \cap I_{hidden} \right|},$$

$$precision = \frac{|\, r_n(u) \cap u^I \cap I_{hidden} \,|}{\left| r_n(u) \cap I_{hidden} \right|},$$

где u^I – множество всех купленных пользователем u товаров из I. Значения этих мер вычисляются для каждого посетителя и усредняются. Эксперимент проводится для разбиения исходной выборки на тестовое и обучающее множество несколько раз, например, 100. Значения мер вновь усредняются. Помимо этого существует возможность отбора признаков для множества I_{hidden} скрытых признаков, процедура разбиения выполняется случайным образом, но необходимо задавать их долю, например, 20%. Идея метода оценки позаимствована из машинного обучения, и носит название скользящего контроля (cross-validation), но в случае рекомендательных систем претерпевает ряд модификаций. Нами использована модификация метода, называемая m-кратный скользящий контроль, идея которого состоит в том, что искомая выборка разбивается на m подмножеств объектов, каждое из которых затем используется в качестве тестовой выборки, а оставшиеся – в качестве обучающей. В дополнение к стандартным способам вычисления точности и полноты мы предлагаем нетрадиционные варианты раскрытия неопределенностей,

возникающих при их оценке. Если $\left|u^I \cap I_{hidden}\right| = 0$, то $recall = 1$. Если $\left|r_n(u) \cap I_{hidden}\right| = 0$ и $u^I = 0$, то $precision = 1$, иначе $precision = 0$.

Эксперименты проведены на наборах данных MovieLens о рейтингах фильмов и синтетических наборах данных, сгенерированных нами.

4. Результаты экспериментов

Мы провели серию экспериментов на наборе данных рейтингов фильмов, в котором представлено 1682 фильма, 943 пользователя, каждый из которых оценил не менее 20 фильмов. Все эксперименты проведены на компьютере с процессором Intel Core 2 Duo 2.0 Гц, 3МБ ОП, под управлением Windows Vista. Алгоритмы реализованы на языке Python 2.6. Мы приводим результаты эксперимента по исследованию зависимости точности и полноты поиска от количества скрываемых признаков для указанных данных (применялся 10-кратный скользящий контроль при фиксированном количестве соседей 10).

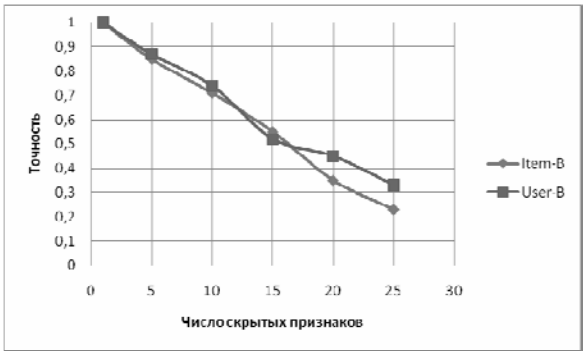

Рис.2. Зависимость точности от числа скрытых признаков

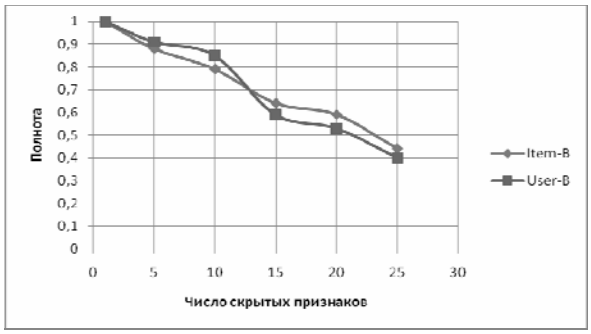

Рис.3. Зависимость полноты от числа скрытых признаков

Как видим по точности поиска методы практически одинаково себя ведут, а по полноте в диапазоне от 1 до 10 скрытых признаков сходство по пользователям работает точнее. Также мы провели аналогичные эксперименты на наборе данных рейтингов фильмов, в которых поэтапно скрывали признаки от 1% до 20%. На графиках для этих экспериментов метод Item-based показывает лучшие результаты только при скрывании малого количества признаков (~1%). В дальнейшем его показатели значительно ухудшаются, тогда как User-based ведет себя достаточно стабильно, более того, начиная с 6-7% полнота начинает возрастать.

Результаты по влиянию количества соседей и размера тестовой выборки мы приводим для синтетического набора данных состоящего из 20 объектов и 20 признаков по диагонали которого расположено четыре прямоугольника из единиц размером 5x5.

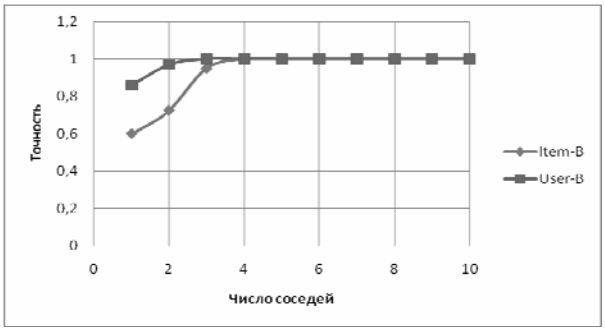

Рис.4. Зависимость точности от числа соседей

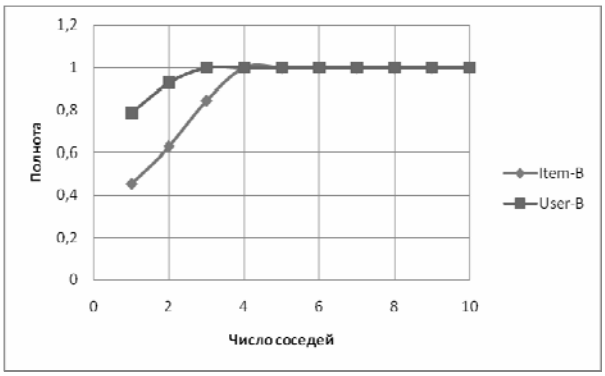

Рис.5. Зависимость полноты от числа соседей

Из этих экспериментов видно, что точность и полнота увеличиваются с ростом количества соседей, причем методу, основанному на сходстве пользователей, требуется меньше соседей, чем на сходстве признаков для достижения такого же качества результатов. Результаты для изменяющегося размера тестового множества носят похожий характер, с ростом размера тестового множества качество предсказаний улучшается, а сходство по пользователям дает лучшие результаты.

5. Заключение

Предложенная методика оценки качества рекомендательных Интернет-сервисов позволяет сравнивать методы рекомендательных систем и настраивать их параметры. В ходе экспериментов было установлено, что классические методы группы user-based ведут себя лучше с точки зрения точности и полноты поиска при количестве скрытых признаков около 10 на реальных данных. Данная методика также позволяет сравнивать любые другие методы поиска рекомендаций, в том числе и основанные на алгоритмах бикластеризации [Ignatov et al., 2008], [Игнатов и др., 2008]. Сравнительные исследования этой группы методов представляют дальнейшее направление развития нашей работы.

Список литературы

[Badrul et al., 2000] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, John Riedl, "Analysis of recommendation algorithms for e-commerce", ACM Conference on Electronic Commerce, pp. 158-167, 2000.

[Golberg et al., 1992] David Goldberg, David A. Nichols, Brian M. Oki, Douglas B. Terry. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM , volume 35(12), 1992, pp. 61-70.

[Сегаран, 2008] Тоби Сегаран. Программируем коллективный разум, СПб.: Символ-плюс, 2008, 368 с.

[Ignatov et al., 2008] Ignatov, Dmitry I., Kuznecov, Sergey Olegovich. Concept-based Recommendations for Internet Advertisement. // Proceedings of the Sixth International Conference on Concept Lattices and Their Applications, 2008. С. 157—166

[Игнатов и др., 2008] Игнатов Дмитрий Игоревич, Кузнецов Сергей Олегович. Методы разработки данных (Data Mining) для рекомендательной системы Интернет-рекламы.// Труды одиннадцатой национальной конференции по искусственному интеллекту с международным участием, 2008. Т. 2. № КИИ-2008. С. 34—42