РЕШЕНИЕ ЗАДАЧИ ПРОФИЛИРОВАНИЯ НА ОСНОВЕ АНАЛИЗА ДАННЫХ

Бусаров В. Г., студент III курса кафедры информационно-аналитических систем СПбГУ, VyacheslavBusarov@gmail.com

Аннотация

В данной работе описан новый подход к решению задачи профилирования объектов по признакам. Будучи актуальной на сегодняшний день, она требует работы с большими объёмами данных и не может решаться вручную. Мы привлекаем методы поиска часто встречающихся наборов для вычислительно эффективного получения качественного результата и приводим оптимальный алгоритм для решения поставленной задачи. В статье предложен способ обобщения исходных данных для оптимизации процесса их обработки и кратко обоснован выбор наилучшего решения подзадачи.

Наиболее значимой ценностью данного исследования является конкретный практический результат, имеющий перспективы внедрения в реальное бизнес-предприятие. Помимо этого, одна из социальных сетей также заинтересовалась результатами данной работы, что ещё раз подчёркивает её актуальность. Эксперименты проводились на "живых" данных ныне существующей компании, с которыми ранее никто не работал.

Постановка задачи

Пусть имеется некоторое количество объектов и упорядоченное множество различных однородных общих для всех признаков. Входные данные задачи представляют из себя совокупность выборок из множества признаков, каждая из которых относится к конкретному объекту и описывает его. Требуется найти набор признаков, наиболее ёмко характеризующих данный объект, то есть достаточно часто встречающихся в его исходных описаниях. Минимальный порог частоты появления набора является входным параметром, так как напрямую влияет на размер каждого профиля. В общем случае это будет полезно для:

- получения краткой и ёмкой характеристики;
- изучения свойств, связанных с профилями объектов;
- получения новых знаний об объекте;
- формирования предположений о дальнейшем поведении или использовании объекта;
- динамического создание сезонных профилей.

Одним из частных случаев данной задачи является проблема профилирования сети организаций торговли, в частности ресторанов, которые мы и будем называть объектами. Признаки в данном случае − это блюда, предлагаемые в меню, а входной набор выборок − транзакции оплаченных счетов. В результате мы получим набор блюд, наиболее популярных в каждом заведении, что поможет решать конкретные задачи: оптимизация закупки сырья, более эффективное управление ценами и ассортиментом, формирование персональных рекомендаций, планирование рекламных компаний (промо-акций).

Обзор существующих решений

В опубликованных по схожей тематике исследованиях[4][5] делается акцент на выделении признаков из текстов на естественном языке с последующим прямым сопоставлением их объектам. Фактически соотносится объект и мнения о нём. Отличие нашей задачи в наличии фиксированного общего словарь признаков, в отсутствии возможности сравнивать объекты между собой до выделения профилей и акцент на выделении свойств объектов.

Описание подзадачи

Для решения проблемы профилирования предлагается воспользоваться алгоритмом поиска часто встречающихся наборов. Он, в свою очередь, является частью решения задачи поиска ассоциативных правил[1]. Давайте введём некоторые понятия и обозначения.

- пусть имеется упорядоченное множество различных однородных признаков $\mathrm{I} = \{i_1, i_2, \dots, i_n\}$;
- входные данные (набор выборок) обозначим как $T^n = \{\tau_1, \tau_2, \dots, \tau_n \mid \tau_i \subseteq \mathrm{I}\}$;
- для каждого набора признаков $\varphi \subseteq \mathrm{I}$, $\varphi(\tau) = 1$, если признаки из φ совместно встречаются в $\tau \in T^n$;
- частота встречаемости (поддержка) φ в T^n − это
$$v(\varphi) = \frac{1}{n}\sum_{i=1}^{n} \varphi(\tau_i);$$
- параметр δ − минимальная поддержка (*MinSupp*), и если $v(\varphi) \geq \delta$, то набор φ частый.

Из всех алгоритмов поиска ассоциативных правил для решения задачи профилирования нам понадобится только поиск частых наборов, удовлетворяющих порогу MinSupp. Будем формировать характеризующий набор признаков $\{i_1, i_2, \dots, i_k\} \subseteq \mathrm{I}$ при некотором k для каждого объекта обособленно, составляя его из элементов найденных в подзадаче. Анализ,

произведённый с учётом взаимосвязи признаков, даст более качественный результат, нежели обособленный подбор. Для того, чтобы наше решение было оптимальным, выберем наиболее вычислительно эффективный алгоритм поиска частых наборов среди имеющихся на сегодняшний день.

Решение задачи

Для простоты восприятия опишем решение на рассматриваемом примере профилирования сети ресторанов, где профилем каждого заведения является набор наиболее популярных блюд. Важным является тот факт, что мы решаем поставленную задачу для каждого объекта в отдельности, несмотря на то, что словарь признаков и входной набор транзакций у них общие. Итоговый алгоритм разбивается на 6 этапов:

I. Фильтрация "пустых" данных

Для того, что бы не работать с информацией, не имеющей никакого смысла, необходимо избавиться от признаков, не описывающих ни один объект. Это очевидным образом осуществляется за линейное время O(N) с использованием O(M) дополнительной памяти, где N − первоначальное количество транзакций, M − количество признаков. В общем случае это легко делается с использованием Хэш-таблиц, в Java это удобнее реализовать с использованием java.util.HashSet<E>, добавляя туда все упомянутые в описаниях признаки, где в итоге из общего множества сохранятся только актуальные.

II. Обобщение с целью оптимизации (устранение избыточности)

Каждая транзакция фактически содержит данные об отдельном клиентском заказе, что является ненужной информацией для формирования профиля заведения. При решении поставленной задачи нам не важно, содержались ли данные блюда в одном чеке, нам важно, что они были заказаны в одном месте в один и тот же день. Результатом IV этапа алгоритма будут наборы вида $\alpha = \varphi \cup y \subseteq I$, $v(\varphi) \geq \delta$. Если построить правила по обобщённым транзакциям, то они будут выглядеть так: "Если φ востребован в данном ресторане, то в тот же день популярен и y." Если это будет происходить в масштабах одного заказа, это никак не повлияет на точность описания объекта, однако данная оптимизация уменьшает количество входных данных на несколько порядков. Таким образом обобщённые данный будут иметь вид, отражённый на рис.1. II шаг осуществляется за время O(N) с использованием O(1) дополнительного пространства.

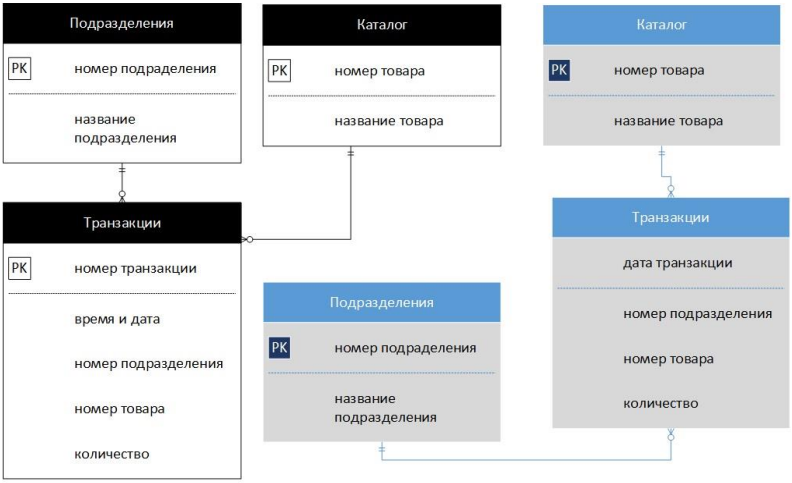

рис.1 Структура входных данных(чёрное), обобщённых (голубое)

III. Фильтрация с целью учёта количества

Большинство алгоритмов поиска частых наборов никак не учитывают количественные характеристики товара. Решения, всё же принимающие во внимание данный показатель, делают это некорректно: если в каком-то отдельно описании признак был несоразмерно востребован, то он тут же попадает в число часто встречающихся, что может заметно размыть итоговый результат. Во избежание этого, следует предварительно исключить непопулярные блюда из каждой обобщённой транзакции. Порог востребованности естественно вычислять следующим образом: отбросив максимальное и минимальное значения в данный день, взять среднее арифметическое среди оставшихся величин[6]. Это объясняется тем, что нам достаточно избавиться только от крайних показателей популярности в текущей транзакции, после чего информация о количестве потеряет значимое влияние на результат. Затраты времени − O(N), затраты памяти − O(1).

IV. Форматирование

На вход алгоритмы поиска частых наборов требуют транзакции как совокупность элементов, входящих в каждую из них. Форматирование происходит очевидным образов за O(N) времени и O(1) памяти.

V. Поиск частых наборов

Можно использовать одни из наиболее вычислительно эффективных алгоритмов PrePost+[2] и Relim[3], сравнительные показатели работы которых указаны в соответствующих статьях. Асимптотика алгоритмов не имеет значения, ввиду использования эвристик.

VI. Выделение профилей

Как уже было отмечено выше, входной параметр MinSupp напрямую влияет на количество признаков в каждом профиле. Переход от часто встречающихся наборов к профилям осуществим пересечением полученных множеств. Регулировать данный параметр будем экспериментально. Сложность − O(M × k) времени и O(M) памяти, где k - количество частых наборов, полученных на этапе V.

Несомненный превосходством приведённого выше алгоритма над простым статистическим методом подсчёта количества товара является анализ признаков в совокупности, вместо обособленного выбора. Также здесь имеется возможность сезонного анализа, когда для получения профиля мы рассматриваем только транзакции конкретного периодичного временного промежутка.

Оптимизации и отсечения на этапах I − III заметно ускоряют работу алгоритма, не влияя на результат, а с учётом выбора на шаге V наименее трудоёмкого подхода, это решение является вычислительно эффективным.

Эксперименты

Некоторой сетью ресторанов были предоставлены данные о работе компании, представляющий из себя три таблицы. Их формат описан на рис.1. Первоначально в таблице "Транзакции" находилось приблизительно 4.9×10^6 записей, описывающих клиентские чеки по всем ресторанам города Москва, чьи названия в статье заменены на условные.

Очевидно, профилирование не имеет смысла при наличии единственного объекта, т.к. одна из мотиваций появления данной задачи − поиск отличительных особенностей объектов. В нашем случае их было 23.

Как уже было отмечено ранее, каждый ресторан анализировался на отдельной итерации, но из-за общего словаря признаков (блюд) мы получаем сопоставляемые ответы. Все алгоритмы реализованы на Java 8.

Для решения подзадачи, поиска частых наборов, мы использовали алгоритм Relim[3], являющийся не только вычислительно эффективным, но и простым в реализации, что отмечает и его автор.

Первоначальное количество транзакций на два порядка выше, чем та же величина после обобщения по датам (из-за большого количества отдельных заказов в каждый из дней). Таким образом, без данной оптимизации поиск частых наборов был бы значительно усложнён

увеличением вычислительной сложности.
Очевидно, что готовое решение можно анализировать с нескольких точек зрения для различных целей. Вот несколько интересных результатов:

1. Список наиболее популярных десертов в каждом из заведений сети. Именно эти лакомства попали в компактные профили объектов, вот некоторые из них.
    - "Новая площадь" − панна котта земляника 150г,
    - "Мэрия" − торт сметанник 900 г,
    - "Кофейня на Рождественке" − торт Французский 1000 г,
    - "ГУМ" − яблочный пирог 1000 г,
    - "Аврора" − сырники ванильные 180г.
2. Кофе пикколо популярно только в ресторане "Комсомольский"
3. В Домодедово наибольшим спросом пользуются разного рода сэндвичи. Можно предположить, что клиенты берут еду на борт самолёта.
4. Во многих местах сети чаще всего заказывают разного рода десерты, в то время как в "Белых садах", "Якиманке" и "Мэрии", судя по набору блюд, люди предпочитают обедать (в профиле практически нет кофе и десертов, в основном сытные блюда).
5. В "Покровке" и "Кофейне на Кудринской" частенько завтракают (каши, омлет, сырники).

Если подойти к выделению конкретных фактов из профилей с других сторон, можно составить наиболее популярные сочетания "напиток − десерт" в отдельных заведениях, подготовить промо-акции и комплексные предложения или сделать что-то подобное.

В процессе анализа были найдены как большие так и компактные профили. Рассмотрев их разность по одному объекту можно получить ещё более неочевидные знания, так как таким образом все тривиальные комбинации будут отброшены. Данные, с которыми мы работали, и готовые профили опубликованы в репозитории *github.com/BusarovVyacheslav/profiling_problem*

Заключение

Итак, в данной работе мы привели новый подход к задаче профилирования на основе поиска частых наборов. Эффективность созданного нами алгоритма определяется

- предварительными обобщениями с целью уменьшения количества данных,

- использованием наиболее вычислительно совершенного алгоритма поиска частых наборов среди всех имеющихся на сегодняшний день,
- эффективным алгоритмом формирования профиля по найденным частым наборам признаков.

Практическим результатом работы стало формирование профилей конкретной организации, посредствам обработки "живых" реальных данных. Этот результат имеет перспективу практического применения в реально существующем предприятии.

Как итоговое решение задачи профилирования, так и наиболее оптимальный алгоритм поиска частых наборов могут быть включены в промышленные пакеты аналитики, такие как Watson IBM. Результаты данного исследования уже привлекли интерес одной из социальных сетей, рассматривавшей применение профилирования для формирования пользовательских рекомендаций.

Заметим, что, имея более широкий период распределения данных, очевидным образом можно формировать сезонные профили, используя приведённый нами алгоритм.

В перспективе мы планируем применить более сложный подход для этапа III "Фильтрации с учётом количества". Также планируется внедрение поиска "пиков" отдельных признаков − промежутков времени, повторяющихся с некоторой периодичностью, в которые признак достаточно часто встречался в описании данного объекта.

Литература

[1] R. Agrawal, T. Imielinski, A. Swami. *Mining Associations between Sets of Items in Massive Databases.* : ACM-SIGMOD Int'l Conf. on Management of Data, 207−216. 1993

[2] Z. H. Deng and S. L. Lv. *PrePost+ : An efficient N-lists-based algorithm for mining frequent itemsets via Children–Parent Equivalence pruning.* : Expert Systems with Applications, 42(10): 5424−5432. 2015

[3] Christian Borgelt. *Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination.* : OSDM, Proceedings of the First International Workshop on Open Source Data Mining: 66. 2005

[4] A.-M. Popescu, O. Etzioni. *Extracting Product Features and Opinions from Reviews.* : Proceedings of HLT-EMNLP: 339−346. 2005

[5] F. Li, C. Han, M. Huang, X. Zhu, Y. Xia, S. Zhang, H. Yu. *Structureaware review mining and summarization.* : Proceedings of the 23rd international conference on computational linguistics: 653-661. 2010

[6] Ефимов А. Н. *Порядковые статистики — их свойства и приложения*. : издательство «Знание»: 40−41. 1980.