Применение методов генетического поиска для задач обработки ассоциативных правил

В. В. Бова[1], С. Н. Щеглов[2], Д. В. Лещанов[3]
Южный федеральный университет
[1]vvbova@yandex.ru, [2]srg_sch@mail.ru, [3]leshanovv2011@yandex.ru

***Аннотация*. Определены основные проблемы обработки неструктурированных данных поиска ассоциативных правил. В рамках статьи для решения поставленных задач предлагается методика оптимизации входных данных на основе методов генетического поиска. Данный подход позволяет осуществить предварительный отбор и последующее разбиение данных на связанные группы, определить эффективные методики их обработки.**

Ключевые слова:анализ данных; ассоциативные правила; алгоритм; генетический поиск; оптимизация

I. Введение

В настоящее время возникают задачи, связанные с необходимостью обработки больших массивов данных с целью поиска новых закономерностей, установления и выявления новых знаний. Для анализа данных широко используются методы и средства искусственного интеллекта [1, 4, 5]. Однако такие методы, как правило, применяются для обработки структурированных данных, представленных в виде массивов, содержащих значения признаков и выходных параметров экземпляров выборки [1, 4–6]. В настоящее время наблюдается переизбыток неструктурированных данных [1, 2], в которых каждая единица хранения не может быть представлена конечным числом признаков (атрибутов). Такие данные представляются в виде последовательностей связанных событий [1–4]. При этом нет чёткого понимания, что является входными данными, а что выходными. Кроме того, размер каждой транзакции (множества событий, произошедших одновременно) не является фиксированным.

В связи с этим актуальными являются задачи:

- сокращения объемов неструктурированных данных путем удаления избыточных транзакций;
- выявления ассоциативных правил, позволяющих извлекать новые знания на основе имеющихся неструктурированных данных;
- построения моделей на основе больших массивов неструктурированных данных для решения практических задач прогнозирования, классификации и кластеризации данных.

Работа выполнена при финансовой поддержке РФФИ, проект № 18-07-00055 -а

Для обработки больших массивов неструктурированных данных и решения указанных задач целесообразно использование новых гибридных подходов, позволяющих эффективно работать с нечеткой или неполной входной информацией, одним из которых, являются методы генетического поиска. В работе представлен интегрированный подход поиска ассоциативных правил и шаблонов [2, 4, 5, 8–10], позволяющий выявлять новые закономерности вида «если условие, то действие» в имеющихся данных и синтезировать на их основе интерпретабельные базы правил, понятные экспертам в прикладных областях.

II. Постановка задочи

Ассоциативным правилом называется выражение вида $A \Rightarrow B$, при этом множества A и B такие, что $A \cap B = \varnothing$. Для автоматического определения таких правил существуют специальные алгоритмы, которые по входному набору множеств строят набор ассоциативных правил, удовлетворяющих определенным ограничениям. Входными данными для таких алгоритмов является набор множеств. Этот набор называется «базовым набором», а каждое множество внутри него – транзакцией.

Пусть $\sigma(A)$ A – количество транзакций в базовом наборе, в которых присутствуют все элементы множества A, а N – общее число транзакций в базовом наборе. Критериями оценки качественного ассоциативного правила выступают такие величины как поддержка и достоверность.

Поддержка – отношение количества транзакций, содержащих как условие, так и следствие к числу транзакций в базовом наборе.

$$\text{sup(b)}=\frac{\sigma(A\cup B)}{N}$$

Достоверность – отношение числа транзакций, содержащих как условие, так и следствие, к числу транзакций, содержащих только условие.

$$\text{conf(b)}=\frac{\sigma(A\cup B)}{\sigma(A)}$$

Для обеспечения процесса синтеза ассоциативных правил необходимо:

- генерирование всех наборов A с уровнем поддержки, не ниже заданного экспертом порогового значения *minsupport(A)*, в результате чего формируются часто встречаемые наборы;
- генерирование всех правил $A \rightarrow B$ с уровнем достоверности, не ниже заданного экспертом порогового значения minconfidence($A \rightarrow B$)

III. Анализ существующих алгоритмов

Ключевым моментом поиска ассоциативных правил является обнаружение частых наборов. Рассмотрим достоинства и недостатки наиболее распространенных алгоритмов решающих данную проблему.

A. APRIORI

Наиболее распространенным алгоритмом поиска ассоциативных правил является алгоритм apriori [5].

Достоинства алгоритма:

- простота;
- быстрое уменьшение числа сгенерированных кандидатов, при установке высокой минимальной поддержки или относительно разреженном базовом наборе.

Недостатки алгоритма:

- многократное сканирование базового набора;
- большое число сгенерированных кандидатов, при слишком большом наборе данных или при слишком низкой поддержке.

Так при применении этого алгоритма к набору, содержащему шаблон из 100 элементов необходимо сгенерировать порядка 2^{100} кандидатов и выполнить над ними все необходимые проверки. Таким образом, алгоритм эффективен только для небольших наборов, либо при высоком уровне минимальной поддержки.

B. FP-GROWTH

Этот алгоритм – один из самых эффективных и позволяет избежать не только затратной процедуры генерации кандидатов, но и многократного сканирования входного набора [6].

Достоинства алгоритма:

- позволяет избежать затратной процедуры генерации кандидатов, характерной для Apriori и Eclat;
- сжатие базового набора в компактную структуру, обеспечивающие быстрое и полное извлечение предметных наборов;
- число сканирования входного набора сокращено до двух;
- размер дерева обычно меньше размера входного набора данных.

Недостатки алгоритма:

- построение дерева – затратная по времени операция;
- в некоторых случаях, вследствие большого числа узлов и связей, размер FP-дерева может намного превышать размер входного набора данных.

C. ECLAT

Алгоритм Eclat [4], на первом шаге своей работы преобразует горизонтальное представление множеств в вертикальное (так называемые TID-множества) и в дальнейшем ведет работу именно с ним.

Достоинства алгоритма:

- поддержка для любого элемента рассчитывается без сканирования базового набора
- число сканирований базового набора сокращено до одного раза.

Недостатки алгоритма:

- TID-множества могут оказаться слишком большими,
- большое число сгенерированных кандидатов, при малом уровне минимальной поддержки.

IV. Модифицированная методика оптимизации входных данных на основе генетического поиска

Основной задачей при обработке неструктурированных данных является снижение размера транзакций и их количества перед выполнением на них какого-либо алгоритма вычисления ассоциативных правил. Для этого предлагается модифицированная методика оптимизации входных данных на основе генетического поиска.

Шаг 1. Сортировка транзакций по определенному критерию. Возможные варианты: удаление из всех транзакций всех элементов a, для которых $\sup(\{a\})<T$, где $T=\alpha N$, α–коэффициент поддержки, либо на усмотрение эксперта, пользователя.

Шаг 2. Исключение из входного набора всех транзакций, содержащих по окончании первого шага меньше двух элементов. Отдельный популярный объект – не является набором.

Шаг 3. Анализ множества данных на эффективность обработки. Происходит преобразование в TID-форму и фильтрация элементов по правилу $\sup(\{a\})\geq T$ для последующей обработке.

Шаг 4. Построение матрицы из оставшихся элементов по следующему правилу:

$$d_{ij} = \begin{cases} 1, sup\left(\{a_i, a_j\}\right) \geq T \\ 0, sup\left(\{a_i, a_j\}\right) < T \end{cases}$$

Шаг 4. Применение генетического поиска к обработке матрицы. Фактически она представляет собой описание неориентированного графа. Если в этом графе существуют отдельные связные подграфы, то элементы их образующие являются отдельными несвязанными множествами. Предлагается использовать модифицированный генетический алгоритм следующего вида.

1. Ввод начальных параметров.
2. Формирование начальной популяции на основе комбинированных механизмов.
3. Построение целевой функции.
4. Реализация элитной селекции.
5. Реализация одного цикла (поколения эволюции) алгоритма.
6. Выполнение специальных операторов с заданной и случайной вероятностью.
7. Производится оценка вновь образованных решений.
8. Из популяции удаляются повторяющиеся хромосомы (комбинации чисел).
9. Если прошло заданное число итераций, то переход к п. 10, иначе переход к п. 2.
10. Конец работы алгоритма.

В данном алгоритме используется следующая методика кодирования альтернативных решений. Длина хромосомы равна числу вершин исследуемого графа. Числовые значения разрядов хромосомы соответствуют номерам вершин графа. Начальная популяция формируется случайным образом, причем различные хромосомы отличаются друг от друга порядком следования вершин.

Выполним теоретическую оценку нижней l_{min} и верхней l_{max} границы числа внутренней устойчивости. Число генов в хромосоме N равно верхней оценке числа внутренней устойчивости l_{max}. Значением гена является номер вершины в графе G. Заполненная часть хромосомы l представляет собой внутренне устойчивое или независимое подмножество, $l \in [l_{min}, l_{max}]$. Так как параметр l определяет длину внутренне устойчивого или независимого подмножества, то его целесообразно использовать для оценки качества хромосомы. Тогда целевой функцией хромосомы, а также критерием оптимизации будет являться l, а целью оптимизации – максимизация функции, т.е. $F(H) = l_H$, $F(H) \rightarrow max$. Далее формируем начальную популяцию хромосом на основе разработанных механизмов. Она представляет собой множество внутренне устойчивых подмножеств, некоторые из которых могут быть и независимыми. Сначала случайным образом формируется множество вершин, мощность которого выбирается случайно в пределах $[l_{min}, l_{max}]$. Проверяется, является ли это множество внутренне устойчивым, если да, то хромосома записывается в начальную популяцию, если нет, то формируется новое подмножество вершин. Процесс продолжается до тех пор, пока не будет сформирована популяция заданного размера. В данном алгоритме используются следующие операторы: модифицированный упорядочивающий одноточечный оператор кроссинговера (ОК); модифицированный оператор мутации; элитный и равновероятный операторы отбора.

Для оценки качества полученных решений применяется следующая процедура. В хромосоме (строке) выбирается первая позиция, и затем данная числовая последовательность просматривается слева направо. На каждом шаге проверяется выполнение заданного условия. Так, при построении независимых подмножеств заданным условием является отсутствие общих ребер между рассматриваемыми вершинами. В результате выполнения данной процедуры будет сформировано одно экстремальное подмножество некоторой длины. Следовательно, качество полученных решений определяется длиной соответствующих им подмножеств. При формировании на некотором шаге подмножества, аналогичного уже существующему, оно исключается из популяции.

Шаг 5. Обработка оптимизированных входных данных алгоритмами Eclat и FP-growth.

V. Оценка эффективности

Для тестирования эффективности предложенной методики фильтрации входных данных использовались два входных набора:

- корпус Retail [7], содержащий данные о покупках людей (кассовые чеки) в магазине. Этот набор содержит 88161 транзакцию, а число различных элементов в ней равно 16469;
- корпус, содержащий множество различных URL. Каждый URL является отдельной транзакцией, а элементом будет являться символ строки. Этот корпус содержит 997451 URL и 9716 различных элементов в нем.

Ниже приведены графики зависимости количества узлов в построенном FP-дереве от коэффициента поддержки (рис. 1.). Обозначения: FP – работа алгоритма без применения генетического поиска; GA – работа алгоритма с использованием методик генетического поиска

Из графиков видно, что предлагаемая методика позволяет сократить объем построенного дерева в десятки, а иногда и в сотни раз, однако этот эффект уменьшается при стремлении коэффициента поддержки α к нулю. Это объясняется тем, что по мере уменьшения α общее число связей в матрице увеличивается и количество несвязанных элементов и отдельных групп уменьшается.

На рис. 2 показаны зависимости времени решения алгоритма FP-Growth от входного набора данных. T_BFS – время работы алгоритма без фильтрации входных данных; T_DFS,T_NNS – время решения с использованием генетического поиска с различными параметрическими настройками алгоритма. Из анализа графиков следует, что при одних и тех же исходных данных качество решений,

полученных с помощью предложенного метода, превосходит качество решений, полученных классическими алгоритмами.

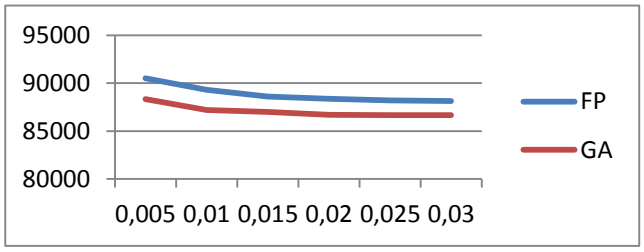

Рис. 1. Зависимость размера числа узлов в дереве от коэффициента поддержки

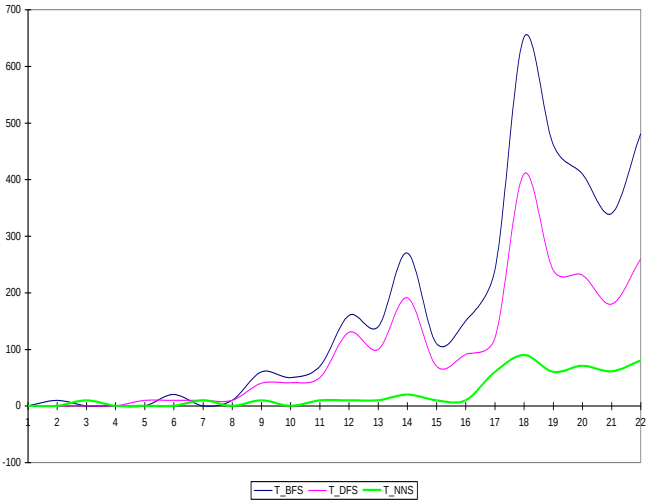

Рис. 2. Зависимость времени решения алгоритма FP-Growth

VI. Заключение

Эксперименты показали, что при решении задачи обработки неструктурированных данных на основе поиска ассоциативных правил использование модифицированных генетических операторов, нестандартных методов поиска и комбинированных моделей позволяет получать набор оптимальных решений. При этом с большой вероятностью среди этих решений может быть найден глобальный экстремум. Из анализа исследованных статистических данных следует, что в общем случае время решения линейно зависит от количества генераций и ВСА приближенно равно $O((nlogn) - (n^3))$.

Список литературы

[1] Карпенко А.П. Современные алгоритмы поисковой оптимизации Алгоритмы, вдохновленные природой: учебное пособие. Москва.: Издательство МГТУ им.Н.Э. Баумана, 2014. 446

[2] Поспелов Д.А. Данные и знания. Искусственный интеллект. В 3 кн. Кн. 1. М: Радио и связь, 1990. 464с.

[3] Shin Y.C. Intelligent systems: modeling, optimization, and control / C.Y. Shin, C. Xu. Boca Raton: CRC Press, 2009. 456 p.

[4] Курейчик В.М., Курейчик В.В. Эволюционные, синергетические и гомеостатические стратегии в искусственном интеллекте: состояние и перспективы. Новости искусственного интеллекта. 2000. № 3. С. 39-67.

[5] Zaki M. Scalable Algorithm for association mining / M. Zaki // IEEE Transactions on Knowledge and Data Engineering. 2000. № 12. С. 372–390.

[6] Agrawal R. Fast algoritms for missing association rules in large databases / R. Agrawal, R. Srikant // Proceedings of the 20th International Conference on Very Large Data Bases. Santiago de Chile. 1994. С. 487–499.

[7] Han J. Mining of frequent patterns without candidate generation: a frequent-pattern tree approach / J. Han, J. Pei, Y. Yin, R. Mao // Data mining and analysis discovery. 2004. Т. 8. № 1. С. 53–87.

[8] Gladkov L.A., Sheglov S.N., Gladkova N.V. The application of bioinspired methods for solving vehicle routing problems. // Procedia Computer Science, 120 (2017). 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017. p. 39–46.

[9] Кулиев Э.В., Шеглов С.Н., Пантелюк Е.А., Логинов О.А. Адаптивный алгоритм стаи серых волков для решения задач проектирования. // Известия ЮФУ. Технические науки, № 7 (192), 2017. Таганрог: изд-во ЮФУ. с. 28–38.

[10] Frequent Itemset Mining Implementations Repository. Retail [Электронный ресурс]. – (URL: http://fimi.ua.ac.be/data/retail.dat/)

[11] Бова В.В., Лещанов Д.В. Семантический поиск знаний в среде функционирования междисциплинарных информационных систем на основе онтологического подхода. // Известия ЮФУ. Технические науки, № 7 (192), 2017. Таганрог: изд-во ЮФУ. с. 80–91.

[12] Borgelt C. An Implementation of the FP□growth Algorithm [Электронный ресурс] / C. Borgelt // Workshop Open Source Data Mining Software. – New York: ACM Press. 2005. – Режим доступа: http://www.osdm.ua.ac.be/ papers/p1-borgelt.pdf