УДК 336.74:004.738.5

АССОЦИАТИВНЫЕ ПРАВИЛА. ИССЛЕДОВАНИЕ АЛГОРИТМА APRIORI НА ПРИМЕРЕ НАБОРА ДАННЫХ INCOME

А.И. Николайчик
Магистрант кафедры математического и информационного обеспечения экономических систем
УО ГрГУ.им. Я.Купалы

Н.В. Марковская
Доцент кафедры математического и информационного обеспечения экономических систем
УО ГрГУ.им. Я.Купалы

УО Гродненский государственный университет имени Янки Купалы
E-mail: nastia036@mail.ru , n.markovskaya@grsu.by

А.И. Николайчик
Окончила Гродненский государственный университет имени Янки Купалы. Магистрант кафедры математического и информационного обеспечения экономических систем УО ГрГУ им. Я. Купалы.

Н.В. Марковская
Доцент кафедры математического и информационного обеспечения экономических систем УО ГрГУ им.Я.Купалы, кандидат физико-математических наук, доцент.

Аннотация. В данной статье рассмотрен такой механизм как ассоциативные правила. Описан алгоритм поиска ассоциативных правил Apriori. Для апробации алгоритма Apriori были использованы анкетные данные, содержащиеся в пакете arules - Income, которые были извлечены Барри Беккером из базы данных переписи 1994 года. Анализ проводился с помощью с помощью программы RStudio с использованием языка R.

Ключевые слова: ассоциативные правила, закономерность, поддержка, достоверность, вероятность, алгоритм Apriori.

Ассоциативные правила представляют собой механизм нахождения логических закономерностей между связанными элементами (событиями или объектами). Пусть имеется A = {a1, a2, a3, …, an} - конечное множество уникальных элементов (list of items). Из этих компонентов может быть составлено множество наборов T (sets of items), т.е. $T \subseteq A$.

Ассоциативные правила A→T имеют следующий вид: если <условие> то <результат>, где в отличие от деревьев классификации, <условие> - не логическое выражение, а набор объектов из множества A, с которыми связаны (ассоциированы) объекты того же множества, включенные в <результат> данного правила. Например, ассоциативное правило если (смородина, тля) то (муравьи) означает, что если на кусте смородины встретилась тля, то ищи поблизости и муравьев.

Понятие «вид элемента a_k» легко может быть обобщено на ту или иную его категорию или вещественное значение, т.е. концепция ассоциативного анализа может быть применена для комбинаций любых переменных.

Выделяют три вида правил:

- полезные правила, содержащие действительную информацию, которая ранее была неизвестна, но имеет логическое объяснение;

- тривиальные правила, содержащие действительную и легко объяснимую информацию, отражающую известные законы в исследуемой области, и поэтому не приносящие какой-либо пользы;

- непонятные правила, содержащие информацию, которая не может быть объяснена (такие правила или получают на основе аномальных исходных данных, или они содержат глубоко скрытые закономерности, и поэтому для интерпретации непонятных правил нужен дополнительный анализ).

Поиск ассоциативных правил обычно выполняют в два этапа:

- в пуле имеющихся признаков A находят наиболее часто встречающиеся комбинации элементов T;

- из этих найденных наиболее часто встречающихся наборов формируют ассоциативные правила.

Для оценки полезности и продуктивности перебираемых правил используются различные частотные критерии, анализирующие встречаемость кандидата в массиве экспериментальных данных. Важнейшими из них являются поддержка (support) и достоверность (confidence). Правило A→T имеет поддержку s, если оно справедливо для s% взятых в анализ случаев:

$$\text{support}(A \rightarrow T) = P(A \cup T)$$

Достоверность правила показывает, какова вероятность того, что из наличия в рассматриваемом случае условной части правила следует наличие заключительной его части (т.е. из A следует T):

$$\text{confidence}(A \rightarrow T) = P(A \cup T)/P(A) = \text{support}(A \rightarrow T) / \text{support}(A).$$

Алгоритмы поиска ассоциативных правил отбирают тех кандидатов, у которых поддержка и достоверность выше некоторых наперед заданных порогов: minsupport и minconfidence. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитику или настолько очевидные, что нет никакого смысла проводить такой анализ. Большинство интересных правил находят именно при низком значении порога поддержки. С другой стороны, низкое значение minsupport ведет к генерации огромного количества вариантов, что требует существенных вычислительных ресурсов или ведет к генерации статистически необоснованных правил.

В пакете arules для R используются и другие показатели - подъемная сила, или лифт (lift), которая показывает, насколько повышается вероятность нахождения T в анализируемом случае, если в нем уже имеется A:

$$\text{lift}(A \rightarrow T) = \text{confidence}(A \rightarrow T) / \text{support}(T)$$

и усиление (leverage), которое отражает, насколько интересной может быть более высокая частота A и T в сочетании с более низким подъемом:

$$\text{leverage}(A \rightarrow T) = \text{support}(A \rightarrow T) - \text{support}(A) \times \text{support}(T)$$

Первый алгоритм поиска ассоциативных правил был разработан в 1993 г. сотрудниками исследовательского центра IBM, что сразу возбудило интерес к этому направлению. Каждый год появлялось несколько новых алгоритмов (DHP, Partition, DIC и др.), из которых наиболее известным остался алгоритм «Apriori» (Agrawal, Srikant, 1994).

Пакет arules позволяет находить часто встречающиеся сочетания элементов в данных (frequent itemsets) и отбирать ассоциативные правила, обеспечивая интерфейс к модулям на

языке C, которые реализуют алгоритмы «Aprior» и «Eclat». Так как обычно обрабатываются большие множества наборов и правил, то для уменьшения объёмов требуемой памяти пакет содержит развитый инструментарий преобразования разреженных входных матриц в компактные наборы транзакций (Hahsler et al., 2016; Огнева, 2012).

Для реализации работы с алгоритмами выделения ассоциаций в arules реализованы специальные типы данных, относящиеся к объектам трех классов: входной массив транзакций (transactions) и на выходе – часто встречающиеся фрагменты данных (itemsets) и правила (rules).

Объекты класса transactions представляют собой специально организованные бинарные матрицы со строками-наборами и столбцами-признаками, содержащие значения элемента 1, если соответствующий признак есть в транзакции, и 0, если он отсутствует. В зависимости от типа данных и способа их загрузки, эти объекты могут иметь разные способы организации и состав дополнительных слотов. В частности, подкласс itemMatrix является одновременно средством представления разреженных матриц с использованием функционала пакета Matrix. Другим способом формирования экземпляров класса transactions является загрузка данных из файла функцией read.transactions().

Поиск ассоциативных правил является не вполне тривиальной задачей, т.к. с ростом числа элементов в A экспоненциально растет число их потенциальных комбинаций. Алгоритм “Apriori” является итерационным, при этом сначала выполняются действия для одноэлементных наборов, затем для 2-х, 3-х элементных и т.д.

На первом шаге первой итерации алгоритма подсчитываются одноэлементные часто встречающиеся наборы. Для этого необходимо пройтись по всему массиву данных и подсчитать для них поддержку, т.е. сколько раз набор встречается в имеющемся наборе данных. При последующем поиске k-элементных наборов генерация претендентов состоит из двух фаз - формирование кандидатов нового уровня на основе $(k-1)$ -элементных наборов, которые были определены на предыдущей итерации алгоритма, и удаление избыточных кандидатов. После того как найдены все часто встречающиеся наборы элементов, выполняют процедуру непосредственного извлечения правил из построенного хеш-дерева.

Результатом анализа транзакций с помощью пакета arules являются объекты класса associations, включающие описания множества отношений между признаками (в виде часто встречающихся фрагментов, или правил), которые отбираются в соответствии с различными перечисленными выше мерами качества. Подкласс rules состоит из двух объектов itemMatrix, представляющих левую lhs (left-hand-side) и правую rhs (right-hand-side) сторону правила $A \rightarrow T$, т.е. A - lhs, T - rhs.

Формирование правил осуществляется функцией apriori() с указанием пороговых значений поддержки и достоверности. Функция summary() обеспечивает частотный анализ правил по их длине и достигнутым мерам качества.

Функция plot() из пакета arulesViz позволяет получать различные формы визуализации синтезированных правил.

Метод «graph» функции plot() показывает правила и составляющие их признаки в виде графа, размер узлов которого пропорционален уровню поддержки каждого представленного правила.

Рассмотрим задачу предобработки и анализа анкетных данных. Для этого возьмём базу Income, содержащуюся в пакете «arules». Данные были извлечены Барри Беккером из базы данных переписи 1994 года.

Загрузим набор данных «Income» (рисунок 1):

```
> data("Income")
> Income
transactions in sparse format with
 6876 transactions (rows) and
 50 items (columns)
```

Рисунок 1. Загрузка набора данных

Это маркетинговые данные содержащие 6876 транзакций по 50 характеристикам. Далее воспользуемся следующей командой:

itemFrequencyPlot(Income,support=0.3,cex.name=0.8)

В результате получим график, представленный на рисунке 2.

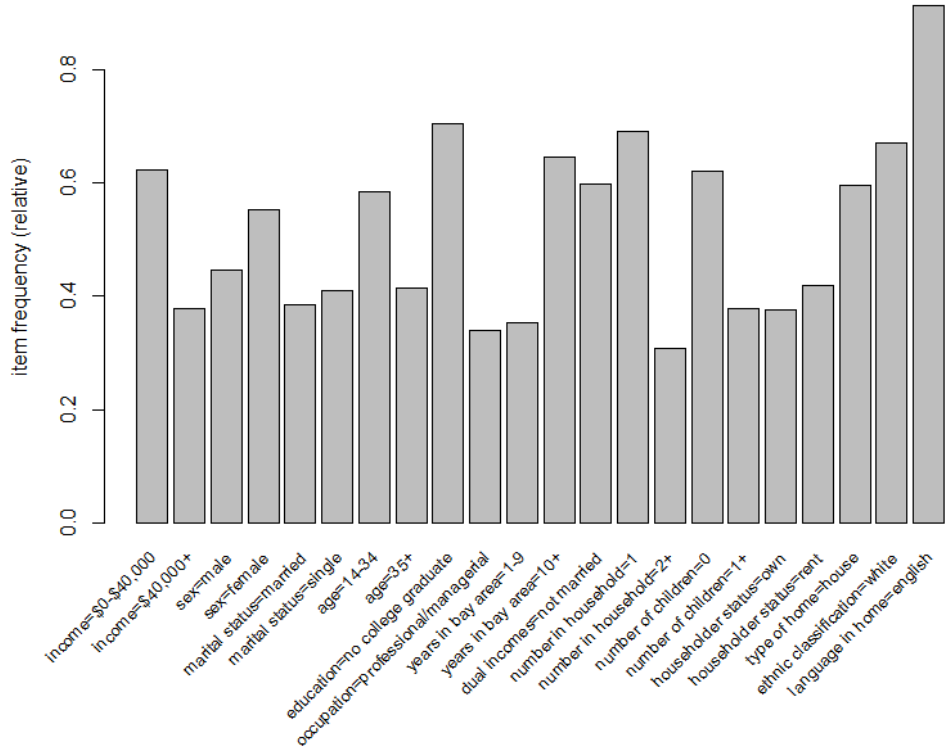

Рисунок 2. Визуализация бинаризованных данных

На данном графике представлены основные характеристики:

– Доход: $0-$40000(Income=$0-$40000), $40000 и более (Income=$40000+);

– Пол: мужской (sex=male), женский(sex=female);

– Семейный статус: женат/замужем (marital status=married), холост/не замужем (marital status=single);

– Возраст: 14-34 (age=14-34), 35 и старше (age=35+);

– Количество детей: нет детей (number of children=0), один и более(number of children=1+);

– Родной язык: английский (language in home=english), и другие.

Для анализа данных используем алгоритм Apriori с минимальной поддержкой 0.1 и значимостью 0.6 (рисунок 3):

```
> rules<-apriori(Income,parameter = list(support=0.1,confidence=0.6))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.6    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 687

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[50 item(s), 6876 transaction(s)] done [0.01s].
sorting and recoding items ... [30 item(s)] done [0.00s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 6 7 8 done [0.09s].
writing ... [17393 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```

Рисунок 3. Использование алгоритма Apriori

В результате работы алгоритм вывел 17393 правил.

Проведем частотный анализ правил по их длине и достигнутым мерам качества (рисунок 4):

```
> summary(rules)
set of 17393 rules

rule length distribution (lhs + rhs):sizes
   1    2    3    4    5    6    7    8
   7  217 1812 5304 6400 3107  538    8

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    4.00    5.00    4.69    5.00    8.00

summary of quality measures:
    support         confidence          lift            count
 Min.   :0.1001   Min.   :0.6000   Min.   :0.8508   Min.   : 688.0
 1st Qu.:0.1108   1st Qu.:0.7052   1st Qu.:1.0710   1st Qu.: 762.0
 Median :0.1254   Median :0.7992   Median :1.2406   Median : 862.0
 Mean   :0.1427   Mean   :0.8028   Mean   :1.3649   Mean   : 981.5
 3rd Qu.:0.1550   3rd Qu.:0.9022   3rd Qu.:1.5096   3rd Qu.:1066.0
 Max.   :0.9129   Max.   :1.0000   Max.   :4.3554   Max.   :6277.0

mining info:
   data ntransactions support confidence
 Income          6876     0.1        0.6
```

Рисунок 4. Частотный анализ правил

Воспользуемся функцией plot() из пакеты aruleViz (рисунок 5):

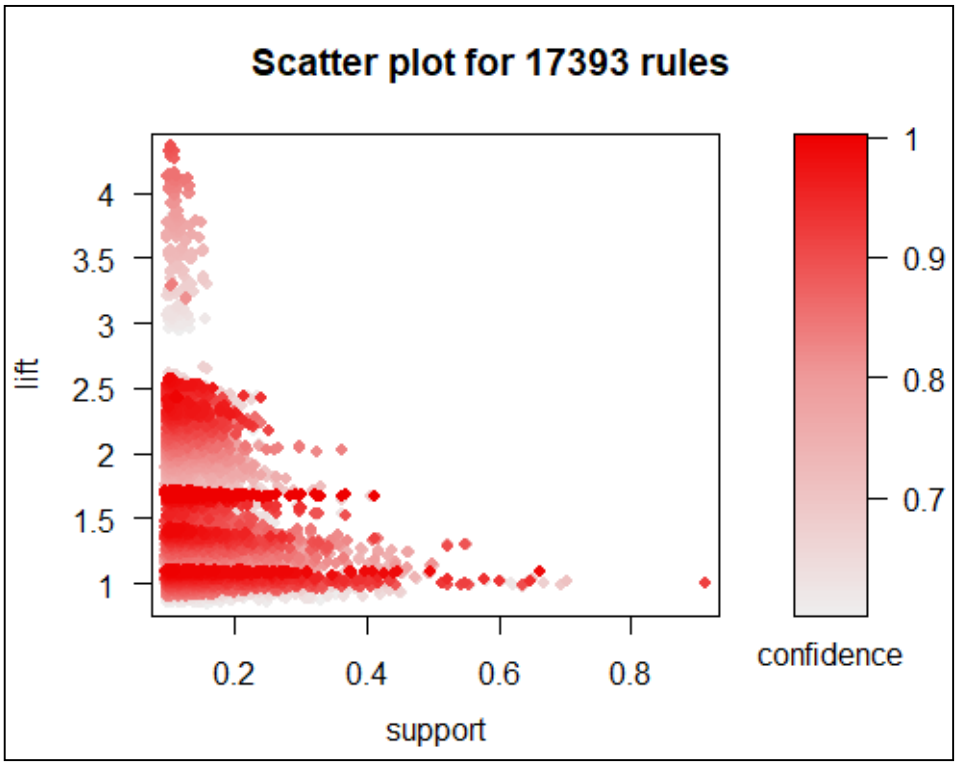

Рисунок 5. Поддержка, лифт и достоверность сгенерированных правил

Для решения задачи выявления характерных особенностей групп населения нас интересуют, в первую очередь, высококачественные правила, имеющие соответствующий признак группы в правой части. Определим, например, признаки группы людей с возрастной категорией 35 и старше (рисунок 6, рисунок 7).

```
> rulesAge35<-subset(rules,subset=rhs %in% "age=35+"&lift>2.13)
> rulesAge35
set of 6 rules
```

Рисунок 6. Определение признаков лиц старше 35 лет

Согласно полученным результатам можно сделать выводы, что для людей старше 35 лет чаще всего встречаются следующие группы:

– женат/замужем, детей нет, в собственности дом;

– этническая классификация – белый, детей нет, тип жилья – дом;

– детей нет, в собственности дом, тип жилья – дом, этническая классификация – белый, родной язык – английский, и др.

С помощью команды plot() построим график шести лучших правил для лиц старше 35 лет в параллельных координатах (рисунок 8).

plot(head(sort(rulesAge35,by = «support»),6),method = «paracoord»)

```
> inspect(head(rulesAge35,n=6,by = "support"))
    lhs                                                  rhs          support confidence     lift count
[1] {marital status=married,
     years in bay area=10+,
     number of children=0,
     householder status=own}                          => {age=35+} 0.1090750  0.8992806 2.168872   750
[2] {years in bay area=10+,
     number of children=0,
     householder status=own,
     type of home=house,
     ethnic classification=white}                     => {age=35+} 0.1089296  0.8853428 2.135257   749
[3] {years in bay area=10+,
     number of children=0,
     householder status=own,
     type of home=house,
     ethnic classification=white,
     language in home=english}                        => {age=35+} 0.1076207  0.8841099 2.132283   740
[4] {marital status=married,
     years in bay area=10+,
     number of children=0,
     householder status=own,
     language in home=english}                        => {age=35+} 0.1039849  0.9005038 2.171822   715
[5] {years in bay area=10+,
     number in household=1,
     number of children=0,
     householder status=own,
     type of home=house,
     ethnic classification=white}                     => {age=35+} 0.1016579  0.8870558 2.139388   699
[6] {years in bay area=10+,
     number in household=1,
     number of children=0,
     householder status=own,
     type of home=house,
     ethnic classification=white,
     language in home=english}                        => {age=35+} 0.1003490  0.8857510 2.136241   690
```

Рисунок 7. Вывод признаков лиц старше 35 лет

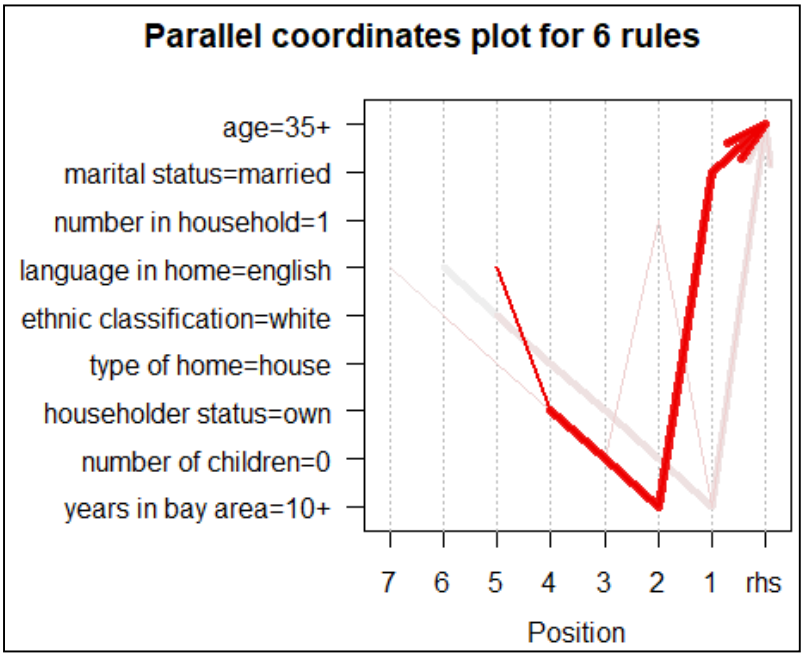

Рисунок 8. График шести лучших правил для лиц старше 35 лет в параллельных координатах

График в параллельных координатах (method=«paracoord») на рисунке 8 показывает, как формируются комбинации признаков правой части при увеличении ее размера, а толщина линий соответствует уровню поддержки.

Построим граф шести лучших правил для лиц старше 35 лет (рисунок 9).

plot(head(sort(rulesAge35,by = «support»),6),method = «graph»,control = list(nodeCol=grey.colors(10),edgeCol=grey(.7),alpha=1))

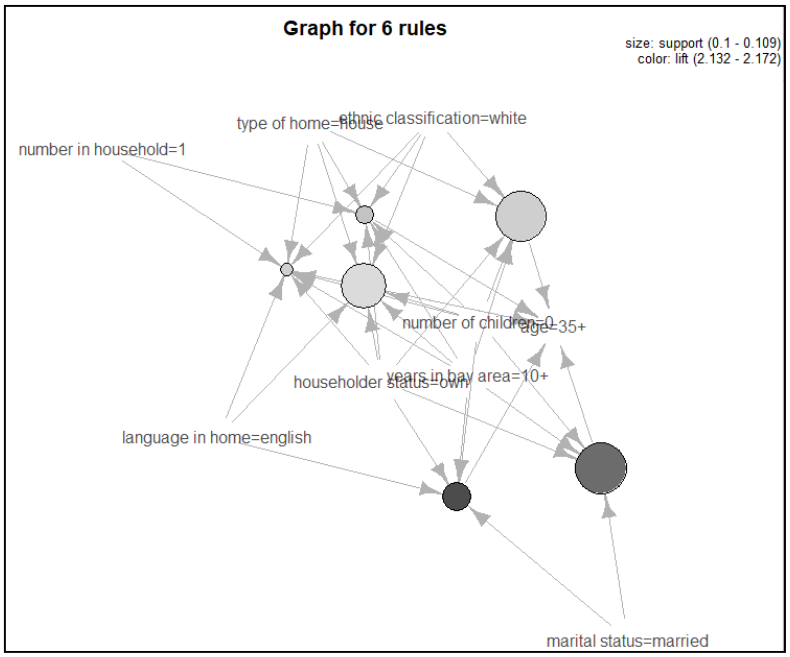

Рисунок 9. Визуализация в форме графа 6 лучших правил для лиц старше 35 лет

Метод «graph» функции plot() показывает правила и составляющие их признаки в виде графа, размер узлов которого пропорционален уровню поддержки каждого представленного правила

Литература:

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

[2]. R. Srikant, R. Agrawal. "Mining Generalized Association Rules", In Proc. of the 21th International Conference on VLDB, Zurich, Switzerland, 1995.

[3]. Шитиков В. К., Мастицкий С. Э./ Классификация, регрессия и другие алгоритмы Data Mining с использованием R, 2017.

ASSOCIATIVE RULES. INVESTIGATION OF ALGORITHM APRIORI ON THE EXAMPLE OF A DATA SET INCOME

A.I. NIKALAICHYK
Master student of the Department of Mathematical and Information Support of Economic Systems,
YKSUG

N.V. MARKOVSKAYA
Associate professor of the Department of Mathematical and Information Support of Economic Systems,
YKSUG

Yanka Kupala State University of Grodno, Republic of Belarus
E-mail: nastia036@mail.ru , n.markovskaya@grsu.by

Abstract. This article describes such a mechanism as association rules. The algorithm of searching for associative rules Apriori is described. For approbation of the algorithm, the personal data contained in the package of arules were used - incomes that were extracted from the 1994 census database. RStudio using R.

Keywords: association rules, regularity, support, reliability, probability, Apriori algorithm.