

УДК: 27.41.41

Применение теории графов для разработки прямого метода построения деревьев решений

М.С. Куприянов¹, Ю.А. Шичкина²¹Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)²Братский государственный университет, ул. Макаренко 40, Братск, Россия. E-mail: strange.y@mail.ru

Статья поступила 7.07.2012, принята 19.11.2012

В статье описываются подходы к построению деревьев решений, основанные на теории графов и матричной алгебре. Эти подходы выгодно отличаются от генетических методов по временной трудоемкости, запасу внутреннего параллелизма и по возможности получения дерева произвольной глубины. Матричный метод является прямым и базируется на операции вычитания матриц. В статье также рассматриваются подходы к регулированию глубины дерева на основе расчета весов всех ветвей. Совокупность ветвей с весом, равным 100%, получаемая с помощью представленного в статье прямого матричного метода, полностью соответствует совокупности ветвей, получаемых с помощью эвристического алгоритма ID3.

Ключевые слова: матрица, метод, генетический метод, деревья решений.

Graph theory application to develop direct method for decision trees constructing

M.S. Kupriyanov¹, Yu.A. Shichkina²¹St. Petersburg State Electrotechnical University "LETI", 5 Professor Popov str., St. Petersburg, Russia²Bratsk State University, 40 Makarenko str., Bratsk, Russia. E-mail: strange.y@mail.ru

The article received 7.07.2012, accepted 19.11.2012

The approaches to decision trees construction based on the graph theory and matrix algebra are described in the article. These approaches compares favourably with the genetic methods as to temporal labour intensiveness, internal parallelism reserve and opportunities to obtain a free intensity tree. The matrix method is direct and based on the matrixes subtraction operation. The article also considers the approaches to the tree intensity control based on the calculation of the weight of all the branches. The total combination of the branches having the weight equal to 100% being obtained by the direct matrix method presented in the article completely corresponds to the total combination of branches being obtained with the help of ID3 heuristic algorithm.

Ключевые слова: matrix, method, genetic method, decision trees.

Введение. Стремительное развитие информационных технологий, в частности прогресс в методах сбора, хранения и обработки данных, позволяют многим организациям собирать огромные массивы данных, которые необходимо анализировать. Объемы этих данных настолько велики, что возможностей экспертов уже не хватает, отсюда постоянно увеличивающийся спрос на методы автоматического исследования данных. Одним из таких методов является построение дерева решений.

Первые идеи создания деревьев решений восходят к работам Ховланда (C.I. Hovland) и Ханта (E.B. Hunt) конца 50-х годов XX века. Однако основополагающей работой, давшей импульс для развития этого направления, явилась книга Ханта, Мэрин (J. Marin) и Стоуна (P.J. Stone) «Experiments in Induction», увидевшая свет в 1966 г.

Дерево решений – это способ представления правил в иерархической последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение.

Область применения деревьев решений в настоящее время широка, но все задачи, решаемые этим аппаратом, могут быть объединены в следующие три класса:

Описание данных. Деревья решений позволяют хранить информацию о данных в компактной форме, вместо них мы можем хранить дерево решений, которое содержит точное описание объектов.

Классификация. Деревья решений отлично справляются с задачами классификации, т. е. отнесения объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения.

Регрессия. Если целевая переменная имеет непрерывные значения, деревья решений позволяют установить зависимость целевой переменной от независимых (входных) переменных. Например, к этому классу относятся задачи численного прогнозирования (предсказания значений целевой переменной).

Общая схема построения дерева принятия решений по тестовым примерам выглядит следующим образом.

- Выбираем очередной атрибут C_i , помещаем его в корень.

- Для всех его значений j :

- Оставляем из тестовых примеров только те, у которых значение атрибута C_i равно j .

- Рекурсивно строим дерево в этом потомке.

Очередной атрибут можно выбирать несколькими способами, и, в зависимости от этого, существует ряд известных алгоритмов, таких как: ID3, где выбор атрибута происходит на основании прироста информации, либо на основании индекса Гини; C4.5, который является улучшенной версией ID3 за счет выбора атрибута на основании нормализованного прироста информации; CART и его модификации и другие.

На практике в результате работы этих алгоритмов часто получаются слишком детализированные деревья, которые при дальнейшем применении дают много ошибок. Это связано с явлением переобучения. Для сокращения деревьев используется отсечение ветвей. Еще один существенный недостаток практического применения алгоритма деревьев решений состоит в том, что он основан на эвристических алгоритмах, таких как алгоритм «жадности», где единственно оптимальное решение выбирается локально в каждом узле. Такие алгоритмы не могут обеспечить оптимальность всего дерева в целом и являются трудоемкими.

Применение графов для построения деревьев решения. Пусть обучаемое множество представляют собой набор объектов $C = C_1, C_2, \dots$. Каждый объект $C_i = x_1, x_2, \dots$ является вектором, где $x_1, x_2, \dots$ представляют собой атрибуты объекта.

Обозначим: m – число атрибутов, n – объем выборки, i – номер атрибута, $i = 1..m$, m_i – число значений i -го атрибута, j – номер значения в атрибуте, $j = 1..m_i$, c_{ij} – количество значений в выборке, соответствующих j -му значению i -го атрибута. Например, если C_1 – первый атрибут и $C_1 = \{\text{sunny, overcast, rain}\}$, то c_{13} – количество в выборке значений «rain». Очевидно, что

$$\sum_{i=1}^m \sum_{j=1}^{m_i} c_{ij} = mn$$

На основе данных об атрибутах объектов и множестве решений S можно построить граф взаимосвязей атрибутов $G(V, E)$, где $V = \{v_1, v_2, \dots, v_p\}$ и $E = \{e_1, e_2, \dots, e_q\}$ соответственно множества вершин и ребер графа. Множество V – совокупность всех значений атрибутов обучаемого множества, $p = \sum_{i=1}^m m_i$, E – совокупность всевозможных в обучаемом наборе связей между атрибутами, $q = \sum_{i=1}^m m_i \left(\sum_{j=i+1}^m m_j \right)$. В полученном графе каждая вершина, соответствующая одному значению своего атрибута, будет связана с каждой вершиной, соответствующей отдельному значению другого атрибута. Граф не будет являться полным и, следовательно, $q < \frac{p(p-1)}{2}$. Пример получаемого графа, изображен на рис. 1.

Преобразуем граф взаимосвязей значений атрибутов во взвешенный граф, добавим в него в качестве

весов ребер количество связей значений i -го и j -го атрибутов.

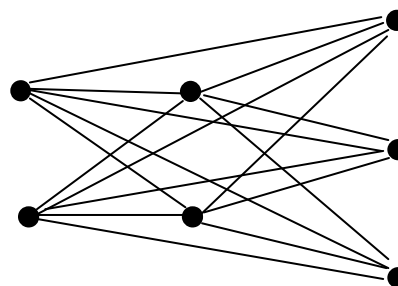


Рис. 1.

Граф взаимосвязей значений атрибутов не является ориентированным и, по своей сути, характеризует многосвязную сложную систему, имеющую много входов и выходов и состоящую из большого числа взаимосвязанных элементов. Следует подчеркнуть неформальность понятия сложной системы, поскольку на современном этапе развития науки нет ее строгого математического определения, охватывающего все интуитивные представления о реальных сложных системах. Если же под сложными системами понимать составной объект, части которого можно рассматривать как системы, закономерно объединенные в единое целое в соответствии с определенными принципами или связанными между собой заданными отношениями, то рассматриваемое множество значений атрибутов является сложной системой.

Систему взаимосвязанных значений атрибутов можно, в соответствии с последним определением, разбить на две подсистемы по принципу их принадлежности тому или иному значению из множества решений S .

Собственно на этом разбиении системы значений атрибутов на несколько подсистем и основан метод построения дерева решений, представляющий собой прямой метод.

Прямой матричный метод построения дерева решений.

1. Пусть число значений множества S равно 2, т.е. $m_s=2$. Обозначим: $s_1=\text{значение1}$, $s_2=\text{значение2}$.

2. Разбить обучаемое множество на два непересекающихся подмножества S_1 и S_2 в соответствии с принципом принадлежности отдельного объекта той или иной группе множества решений S .

3. Построить взвешенный граф взаимосвязей значений атрибутов для каждого подмножества.

4. Построить матрицу смежности для каждого графа. Так как матрица будет симметричной относительно главной диагонали, то можно изначально строить верхнюю треугольную матрицу смежности.

5. На главной диагонали матрицы смежности разместит значения c_{ij} .

6. Рассчитать приоритетность атрибутов в соответствии с формулами энтропии и $Gain$ -индекса:

7.

$$Entropy(S) = - \sum_{i=1}^2 p_i \log_2 p_i ;$$

$$Gain(S, C_i) = Entropy(S) - \sum_{i=1}^2 \frac{|S_i|}{|S|} Entropy(S_i) .$$

8. Построить для подмножеств S_1 и S_2 матрицы смежности **T1** и **T2**.

9. Если в какой-либо из двух матриц смежности **Tk**, где $k=1,2$ существует нулевая строка, то можно сформулировать правило $C_i \rightarrow s_k$.

10. Провести поэлементное сравнение матриц **T1** и **T2**. Если $T1_{ij} = 0$, а $T2_{ij} > 0$, то можно сформулировать правило: $C_k, C_r \rightarrow s_2$, где C_k – атрибут и его значение, соответствующие i -й строке, а C_r – атрибут и его значение, соответствующие j -му столбцу матрицы смежности множества S .

11. Если $T2_{ij} = 0$, а $T1_{ij} > 0$, то можно сформулировать правило: $C_k, C_r \rightarrow s_1$, где C_k – атрибут и его значение, соответствующие i -й строке, а C_r – атрибут и его значение, соответствующие j -му столбцу матрицы смежности множества S .

12. Если в i -й строке несколько элементов, удовлетворяющих условию 9 или 10, то выбирается атрибут с наибольшим *Gain*-индексом.

Регулирования глубины дерева. В теории построения деревьев решений существует понятие регулирования глубины дерева как техники, которая позволяет уменьшать размер дерева решений, удаляя участки дерева, которые имеют маленький вес или низкую вероятность адекватности решения.

Один из вопросов, который возникает в эвристических алгоритмах дерева решений – это оптимальный размер конечного дерева. Так, небольшое дерево может не охватить ту или иную важную информацию о выборочном пространстве. Тем не менее, трудно сказать, когда алгоритм должен остановиться, потому что невозможно спрогнозировать, добавление какого узла позволит значительно уменьшить ошибку. Эта проблема известна как «эффект горизонта». Тем не менее, общая стратегия ограничения дерева сохраняется, т. е. удаление узлов реализуется в случае, если они не дают дополнительной информации.

В нашем случае для решения задачи глубины дерева можно рассчитать вес каждой ветви по формуле:

$$p_i = \frac{T_{ij}}{C_{ij}} 100\%$$

где **T** – матрица смежности множества S . Следует отметить, что совокупность ветвей с весом, равным 100 %, получаемая с помощью описанного прямого матричного метода, полностью соответствует совокупности ветвей, получаемых с помощью эвристического алгоритма ID3.

Тестирование. Для иллюстрации метода возьмем тестовое обучающее множество S , представленное в таблице 1.

По методу ID3 для данного обучающего набора с помощью программы, написанной на языке Fortran была получена совокупность правил:

Read 14 cases (5 attributes).

Decision Tree:

Outlook: Sunny, Humidity: Normal $\rightarrow$ Yes

Outlook: Sunny, Humidity: High $\rightarrow$ No

Outlook: Rain, Windy: Strong $\rightarrow$ No

Outlook: Rain, Windy: Weak $\rightarrow$ Yes

Outlook: Overcast $\rightarrow$ Yes

Таблица 1

Outlook	Temperature	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

В соответствии с алгоритмом разобьем множество решений S на два подмножества:

Таблица 2

S1:

Outlook	Temperature	Humidity	Wind	Play ball
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

Таблица 3

S2:

Outlook	Temperature	Humidity	Wind	Play ball
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Rain	Cool	Normal	Strong	No
Sunny	Mild	High	Weak	No
Rain	Mild	High	Strong	No

Этим множествам будут соответствовать матрицы смежности:

Таблица 4

Матрица смежности T1

	Yes									
	O.S	O	R	T.H	M	C	H.H	H.N	W	W.S
O.S	2	0	0	0	1	1	0	2	1	1
O	0	4	0	2	1	1	2	2	2	2
R	0	0	3	0	2	1	1	2	3	0
T.H	0	0	0	2	0	0	1	1	2	0
M	0	0	0	0	4	0	2	2	2	2
C	0	0	0	0	0	3	0	3	2	1
H.H	0	0	0	0	0	0	3	0	2	1
H.N	0	0	0	0	0	0	0	6	4	2
W	0	0	0	0	0	0	0	0	6	0
W.S	0	0	0	0	0	0	0	0	0	3

После поэлементного сравнения матриц смежности с учетом правил описанного выше матричного метода получим следующий набор правил и рассчитанных для них весов:

Матрица смежности T2

	No									
	O.S	O	R	T.H	M	C	H.H	H.N	W	W.S
O.S	3	0	0	2	1	0	3	0	2	1
O	0	0	0	0	0	0	0	0	0	0
R	0	0	2	0	1	1	1	1	0	2
T.H	0	0	0	2	0	0	2	0	1	1
M	0	0	0	0	2	0	2	0	1	1
C	0	0	0	0	0	1	0	1	1	0
H.H	0	0	0	0	0	0	4	0	2	2
H.N	0	0	0	0	0	0	0	1	0	1
W	0	0	0	0	0	0	0	0	2	0
W.S	0	0	0	0	0	0	0	0	0	3

Первые пять из полученных правил имеют вес, равный 100 %, и соответствуют результатам алгоритма ID3.

Таблица 6

Правила	$T(I,j)$	C_{ij}	Вес
Outlook: Overcast → Yes	4	4	100
Outlook: Sunny, Humidity: High → No	3	3	100
Outlook: Sunny, Humidity: Normal → Yes	2	2	100
Outlook: Rain, Windy: Weak → Yes	3	3	100
Outlook: Rain, Windy: Strong → No	2	2	100
Temperature: Hot, Humidity: Normal → Yes	1	3	33,3
Temperature: Hot, Windy: Strong → No	1	3	33,3
Temperature: Mild, Humidity: Normal → Yes	2	4	50
Temperature: Cool, Windy: Strong → Yes	1	3	33,3
Humidity: Normal, Windy: Weak → Yes	4	6	66,7

Выводы. Прямой матричный метод построения дерева решений позволяет получать те же результаты, что и наиболее известные на сегодняшний день эвристические алгоритмы.

Совокупность правил, получаемых с помощью матричного метода, является более широкой по сравнению, к примеру, с методом ID3, но может быть отрегулирована с помощью несложных расчетов.

Матричный метод предпочтительнее эвристических алгоритмов за счет более низкой трудоемкости. В общем случае трудоемкость эвристических алгоритмов равна $O(m^2k^2)$, где m – число атрибутов, k – наибольшее значение вариантов значений в атрибуте, r – число итераций по построению генов, трудоемкость матричного метода – $O(m^2k^2)$.

Литература

1. Шичкина Ю.А., Рудишин И.А. Расширение программных возможностей Веб-приложения «Tester» // Труды Братского государственного университета. Сер. Естественные и инженерные науки – развитию регионов Сибири. 2012. Т. 1. С. 38-
2. Шичкина Ю.А. Временная сложность алгоритмов. Естественные и инженерные науки - развитию регионов: материалы межрег. НТК. Братск: БрГТУ, 2004. 230 с.

References

1. Шичкина Ю.А., Рудишин И.А. Расширение программных возможностей Веб-приложения «Tester» // Труды Братского государственного университета. Сер. Естественные и инженерные науки – развитию регионов Сибири. 2012. Т. 1. С. 38-
2. Шичкина Ю.А. Временная сложность алгоритмов. Естественные и инженерные науки - развитию регионов: материалы межрег. НТК. Братск: БрГТУ, 2004. 230 с.