# К ВОПРОСУ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО ДЕРЕВА РЕШЕНИЙ

**Виталий Величко**

***Аннотация:*** *В работе приведена формальная постановка задачи построения оптимального дерева решений в терминах задачи бинарной идентификации. Оптимальное дерево решений определено как дерево минимального размера и способное без ошибок классифицировать все объекты из обучающей выборки. Рассматривается случай, когда все атрибуты объектов являются номинальными. Для отбора наилучших правил используется мера информационного выигрыша на основе вычисления условной энтропии. В работе показано, что для задачи построения оптимального дерева решений, сформулированной в терминах задачи бинарной идентификации, существует полиномиальный алгоритм ее решения при условии определения стоимости теста (логического правила) как функции свойств теста. Вычислительная сложность приведенного алгоритма ограничена полиномом третьей степени мощности множества объектов обучающей выборки. Для упрощения рассуждений принято, что для каждого значения целевого атрибута существует не менее одного теста, условная энтропия которого на множестве объектов из обучающей выборки равна 0. Задача построения оптимального дерева решений не является NP-полной задачей при условии задания ограничений на определение функции стоимости логического правила (теста).*

***Ключевые слова:*** *задача бинарной идентификации, оптимальное дерево решений, вычислительная сложность алгоритма.*

***ITHEA Keywords****: F.2.2 [Analysis of algorithms and problem complexity]: Nonnumerical Algorithms and Problems, I.2.6 [Artificial intelligence]: Learning, I.2.4 [Artificial intelligence]: Knowledge Representation Formalisms and Methods.*

## Введение

Деревья решения являются популярным подходом к решению задач Data Mining [Субботин, 2019]. Они позволяют получить иерархическую структуру классифицирующих правил, которая имеет вид дерева [Quinlan, 1986]. Деревья решений могут оценивать значения категориальных (номинальных) атрибутов, имеющих конечное число дискретных значений, а также количественных атрибутов. Древовидные модели, в которых целевая переменная может принимать дискретный набор значений, называются деревьями классификации; в этих древовидных структурах листья представляют метки значений целевого атрибута, а ветви представляют конъюнкцию значений нецелевых атрибутов, которые ведут к этим меткам значений целевого атрибута. Деревья решений, где целевая переменная может принимать непрерывные значения (обычно действительные числа), называются деревьями регрессии [Xindong at all, 2008].

Формально дерево решений можно определить как способ построения классификационной или регрессионной модели в виде древовидной структуры. Узлы дерева подразделяются на решающие узлы (в которых представлены правила) и листья - узлы, дающие решения. Под правилом понимается логическая конструкция, представленная в виде "ЕСЛИ... ТО..." ("IF-THEN").

В процессе обхода дерева в каждом узле в зависимости от проверяемого условия принимается определенное решение – перемещение по той или иной ветке дерева от корня к «листьевым» (конечным) вершинам. В «листьевой» вершине дерева содержится искомое значение интересующего атрибута. В узлах бинарных деревьев решений

ветвление идет только в двух направлениях, т.е. существует только 2 ответа на поставленный вопрос: «да» и «нет». Обучение деревьев решений выполняется индуктивно на основе прецедентов – наблюдений за состоянием моделируемого объекта или процесса. Рассмотрим пример дерева решений, полученного на основе анализа таблицы «Объект-свойство» (Таблица 1). Задача, состоит в том, чтобы на основе анализа примеров ситуаций из таблицы (Таблица 1) сформировать в явном виде правила определяющие значения целевого атрибута (играть в гольф или не играть) в зависимости от значений нецелевых атрибутов (предикторов). Названия и значения атрибутов в какой-то мере условны и служат главным образом для иллюстрации построения и использования деревьев решений.

***Таблица 1. Пример таблицы «Объект-свойство» играть в гольф***

| | Атрибуты (Предикторы) | | | | Целевой атрибут (Target) |
|---|---|---|---|---|---|
| Наименования атрибутов | Наблюдение | Температура | Влажность | Ветер | Играть в гольф |
| Примеры объектов с известными значениями атрибутов | Дождь | Жарко | Высокая | Нет | Нет |
| | Дождь | Жарко | Высокая | Да | Нет |
| | Пасмурно | Жарко | Высокая | Нет | Да |
| | Солнечно | Умеренная | Высокая | Нет | Да |

| | | | | | |
|---|---|---|---|---|---|
| | Солнечно | Прохладная | Нормальная | Нет | Да |
| | Солнечно | Прохладная | Нормальная | Да | Нет |
| | Пасмурно | Прохладная | Нормальная | Да | Да |
| | Дождь | Умеренная | Высокая | Нет | Нет |
| | Дождь | Прохладная | Нормальная | Нет | Да |
| | Солнечно | Умеренная | Нормальная | Нет | Да |

Извлеченные из таблицы правила представлены в виде дерева на Рисунке 1. Сами правила в дизъюнктивной нормальной форме (ДНФ) приведены в Таблице 2. Правила из таблицы получены с помощью аналитической платформы Deductor 4.3 [Deductor, 2020]. Для каждого правила вычислена поддержка (количество примеров из таблицы «объект-свойство» рис.1 для которых правило выполняется) и достоверность (количество примеров для которых выполнение правила дает правильный результат).

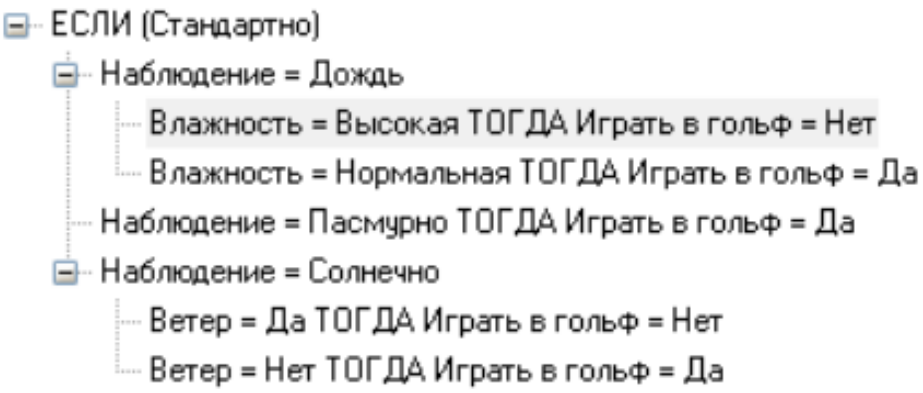


Рисунок 1. Дерево решений играть в гольф

***Таблица 2. Классифицирующие правила играть в гольф***

| N | Условие | Следствие (Целевой атрибут: **Играть в гольф**) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | Наблюдение = Дождь **И** Влажность = Высокая | Нет | 30,00 | 3 | 100,00 | 3 |
| 2 | Наблюдение = Дождь **И** Влажность = Нормальная | Да | 10,00 | 1 | 100,00 | 1 |
| 3 | Наблюдение = Пасмурно | Да | 20,00 | 2 | 100,00 | 2 |
| 4 | Наблюдение = Солнечно **И** Ветер = Да | Нет | 10,00 | 1 | 100,00 | 1 |
| 5 | Наблюдение = Солнечно **И** Ветер = Нет | Да | 30,00 | 3 | 100,00 | 3 |

## Принцип построения дерева решений

Принцип построения дерева [Quinlan, 1986] следующий. Дерево строится «сверху вниз» от корня. Начинается процесс с определения, какой атрибут (предиктор) следует выбрать для проверки в корне дерева. Для этого каждый атрибут исследуется на предмет, насколько хорошо он классифицирует набор данных (разделяет множество примеров на группы по одинаковым значениям целевого атрибута). Когда разделяющий атрибут выбран, для каждого его значения создается ветка дерева, набор данных разделяется в соответствии со значением к каждой ветке, процесс повторяется рекурсивно для каждой ветки. Можно сформулировать правило для выбора атрибута следующим образом: выбранный атрибут должен разбивать множество объектов так, чтобы получаемые в итоге подмножества состояли из объектов, имеющих одно значение целевого атрибута, или были максимально приближены к этому, т.е. количество объектов, которые имеют другие значения целевого атрибута в каждом из этих множеств было как можно меньше. Также проверяется заданный критерий остановки ветвления дерева, который позволяет ограничить выделение детализированных и малозначащих правил. Целью применения критериев остановки ветвления является выделение наиболее полных и точных правил. Под термином наиболее полные правила будем понимать правила, которые истинны для наибольшего количества объектов, имеющих одинаковое значение целевого атрибута. Полнота правила характеризуется его относительной поддержкой. Под термином наиболее точные правила будем понимать правила, которые истинны только для одного значения целевого атрибута. Точность правила характеризуется его относительной достоверностью.

Алгоритмы построения дерева решения, использующие приведенный принцип, такие как ID3, C4.5 являются «жадными». На каждой итерации алгоритма, которая состоит в выборе разделяющего атрибута, максимизируется определенный «локальный» критерий оптимальности в предположении, что получившееся дерево в целом будет «оптимальным». Существуют различные критерии выбора атрибута для расщепления

дерева. Наиболее известные – «мера информационного выигрыша» (англ. information gain measure, gain ratio) или мера энтропии и индекс Gini. При помощи индекса Gini атрибут выбирается на основании расстояний между распределениями значения целевого атрибута. Часто алгоритмы построения деревьев решений дают слишком детализированные деревья, которые имеют много узлов и ветвей. Это связано с явлением переобучения, для избежания которого используется алгоритм отсечения ветвей (pruning). Также для устранения этого недостатка используют метод комитетов из решающих деревьев [Ho,1998]. Популярность использования деревьев решений связана с наглядностью и и возможностью получения правил в явном виде. Алгоритмы построения деревьев решений реализуют наивный принцип последовательного просмотра атрибута. Но дерево решений принципиально не способно находить наиболее полные и точные правила в данных. Приведем простой пример, иллюстрирующий это утверждение. Дана таблица «объект-свойство» (Таблица 3), содержащая 8 примеров, 2 значения целевого атрибута, каждое из которых представлено 4 примерами.

***Таблица 3. «Объект-свойство» пример 2***

| Атрибуты / Объекты | a0 | a1 | a2 | Target |
|---|---|---|---|---|
| 1 | FALSE | FALSE | A | t0 |
| 2 | FALSE | FALSE | A | t0 |
| 3 | TRUE | TRUE | D | t0 |

| Атрибуты / Объекты | a0 | a1 | a2 | Target |
|---|---|---|---|---|
| 4 | TRUE | TRUE | E | t0 |
| 5 | FALSE | TRUE | B | t1 |
| 6 | FALSE | TRUE | C | t1 |
| 7 | TRUE | FALSE | F | t1 |
| 8 | TRUE | FALSE | F | t1 |

Очевидно, что наилучшее дерево решений для приведенного примера содержит 4 правила и имеет вид, приведенный в Таблице 4. Необходимо уточнить, что для значения целевого атрибута t0 вместо правила №1 может быть также сформировано правило a2 = A, а для значения целевого атрибута t1 вместо правила №4 может быть также сформировано правило a2 = F. Данные правила по критерию меры информационного выигрыша эквивалентны правилам №1 и №4 и общее количество правил (четыре) в наилучшем дереве решений все равно не изменяется.

***Таблица 4. Оптимальные классифицирующие правила пример 2***

| N | Условие | Следствие (Целевой атрибут, Target) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | a0 = FALSE **И** a1 = FALSE | t0 | 25,00 | 2 | 100,00 | 2 |
| 2 | a0 = TRUE **И** a1 = TRUE | t0 | 25,00 | 2 | 100,00 | 2 |
| 3 | a0 = FALSE **И** a1 = TRUE | t1 | 25,00 | 2 | 100,00 | 2 |
| 4 | a0 = TRUE **И** a1 = FALSE | t1 | 25,00 | 2 | 100,00 | 2 |

“Жадные” алгоритмы построения дерева решения не выбирают атрибуты a0 и a1 для разделения множества примеров по значениям целевого атрибута (Таблица 3), потому что мера информационного выигрыша для любого из атрибутов a0 или a1 равна 0 (каждое из 2 значений атрибутов a0 и a1 равновероятно для обеих значений целевого атрибута t0 и t1, которые также равновероятны). Для построения дерева решений всегда будет выбран атрибут a2. Правила в ДНФ, полученные с помощью "жадных" алгоритмов построения дерева, приведены в Таблице 5.

***Таблица 5. Классифицирующие правила при использовании "жадных" алгоритмов пример 2***

| N | Условие | Следствие (Целевой атрибут, Target) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | a2 = A | t0 | 25,00 | 2 | 100,00 | 2 |
| 2 | a2 = B | t1 | 12,50 | 1 | 100,00 | 1 |
| 3 | a2 = C | t1 | 12,50 | 1 | 100,00 | 1 |
| 4 | a2 = D | t0 | 12,50 | 1 | 100,00 | 1 |
| 5 | a2 = E | t0 | 12,50 | 1 | 100,00 | 1 |
| 6 | a2 = F | t1 | 25,00 | 2 | 100,00 | 2 |

Всего выделено 6 правил, из которых 4 имеют поддержку 1, т.е. справедливы только для одного объекта и не обладают обобщающей способностью.

## Вычислительная сложность построения оптимальной процедуры бинарной идентификации

В общем случае, при использовании "жадных" алгоритмов, задача построения оптимального дерева решений NP-полная [Hyafil and Rivest, 1976]. Это очевидно, потому что для таких алгоритмов локальные оптимальные решения не могут гарантировать получение глобального оптимального дерева решений. NP-полнота задачи построения оптимального дерева решений рассматривается в работах [Garey, 1972], [Hyafil and Rivest, 1976]. В [Hyafil and Rivest, 1976] приведена обобщенная формулировка задачи бинарной идентификации, без задания каких-либо ограничений на множество бинарных тестов и свойств элементов множества бинарных тестов. Под оптимальным бинарным деревом понимается дерево, которое минимизирует ожидаемое количество тестов, необходимых для идентификации неизвестного объекта. Авторами доказывается, что в общем случае построение оптимальных бинарных деревьев решений является NP-полной задачей. Также утверждается, что эта модель идентична модели, изученной в [Garey, 1972]. Рассмотрим более подробно данную модель, и приведем формальную постановку задачи бинарной идентификации, которая была предложена в [Garey, 1972]. Дано:

a) конечное множество из $n$ объектов, $O_1,O_2,...,O_n$ , которые являются возможными для некоторого неизвестного объекта;

b) соответствующее множество $n$ известных вероятностей, $p_1,p_2,...,p_n$ , удовлетворяющих $0 < p_i \le 1$ , где $p_i$ - вероятность того, что неизвестный объект является $Q_i$ , и поскольку мы рассматриваем только один неизвестный объект – $\sum_{i=1}^{n} p_i = 1$;

c) конечное множество из $m$ бинарных тестов или вопросов, $Q_1,Q_2,...,Q_m$ , каждый из которых, является функцией $Q_j : \{O_1,O_2,...,O_n\} \to \{TRUE,FALSE\}$ ,

где $Q_j(O_i)$ определяет результат применения теста $Q_j$ для неизвестного объекта $O_i$;

d) соответствующее множество из $m$ стоимостей $C_1,C_2,...,C_m$, где $C_j$ затраты на выполнение теста $Q_j$, каждое $C_j > 0$.

Табличная форма описания задачи приведена в таблице 6. В столбцах приведены объекты, в строках тесты, а ячейках – результат применения теста $Q_j$ к объекту $O_i$ – $Q_j(O_i)$.

***Таблица 6. Табличная форма описания задачи бинарной идентификации***

| | | $p_1$ | $p_2$ | … | $p_n$ |
|---|---|---|---|---|---|
| | | $O_1$ | $O_2$ | … | $O_n$ |
| $C_1$ | $Q_1$ | *TRUE* | *FALSE* | … | *TRUE* |
| $C_2$ | $Q_2$ | *FALSE* | *FALSE* | … | *TRUE* |
| … | … | … | … | … | … |
| $C_m$ | $Q_m$ | *FALSE* | *TRUE* | … | *FALSE* |

Стоимость процедуры бинарной идентификации будет определяться как ее средняя стоимость, рассчитанная по всем тестам как сумма произведения стоимости каждого теста на вероятность того, что тест будет использован при применении процедуры бинарной идентификации. Эквивалентно, среднюю стоимость можно вычислить как сумму по всем объектам произведения вероятности объекта, на сумму затрат на вопросы, которые задаются, для идентификации объекта, когда он является неизвестным объектом. Тогда оптимальная процедура бинарной идентификации - это такая, которая обеспечивает минимальную стоимость всех процедур для одной и той же проблемы. В [Garey, 1972] доказывается, что вычислительная сложность построения оптимальной процедуры бинарной идентификации с помощью алгоритма обратной индукции пропорциональна to $m \cdot n \cdot 2^n$ , т.е. оптимальная процедура бинарной идентификации является NP-полной задачей.

## Формальная постановка задачи построения оптимального дерева решений решений

Рассмотрим формальную постановку задачи построения оптимального дерева решений на основе анализа таблицы «объект-свойство» в терминах задачи бинарной идентификации. Рассмотрим случай, когда все атрибуты имеют конечное число дискретных значений.

Обозначим:

$V$ – конечное множество значений целевого атрибута (target) $V = \left\{ v_j \middle| j = \overline{1, n_1} \right\}$, $n_1 = |V|$ – количество значений целевого атрибута;

$W$ – конечное множество значений нецелевых атрибутов (предикторов) $W = \bigcup_{i=1}^{n_2} W^i$ , $n_2$ – количество нецелевых атрибутов, $W^i$ – множество

значений *i*-ого нецелевого атрибута, $W^i = \left\{ r_j^i \middle| j = \overline{1,n_3} \right\}$, где $r_j^i$ – *j*-тое значение *i-г*о атрибута, $n_3$ – количество значений *i-г*о нецелевого атрибута;

*A* – конечное множество известных объектов $A = \left\{ a_k \middle| k = \overline{1,n_4} \right\}$, где $n_4$ – количество известных объектов. Для каждого объекта $a_k \in A$ определено только одно значение $v_{a_k} \in V$ и множество $R_{a_k} = \left\{ r_j^i \middle| r_j^i \in W^i, j \in \{1...n_3\}, i = \overline{1,n_5} \right\}$, где $r_j^i$ – *j*-тое значение *i-г*о нецелевого атрибутов, $n_5$ – количество нецелевых атрибутов, которые определены для объекта $a_k$. Иначе, каждый объект $a_k$ можно представить в синтаксисе логики высказываний следующей формулой $a_k = \bigvee_{i=1}^{n5} r_j^i$, которая определяет имликацию: $\bigvee_{i=1}^{n5} r_j^i \to v_{a_k}$. Для каждого значения целевого атрибута $v_j$ известно множество $V_j = \left\{ a_k \middle| k = \overline{1,n_6} \right\}$, где $n_6$ – количество объектов из множества *A*, имеющих значение целевого атрибута $v_j$. Для каждого $v_j$ определено $p_j = \frac{|V_j|}{|V|}$ – вероятность наблюдения значения целевого атрибута $v_j$, $0 < p_j \le 1$, $\sum_{j=1}^{n_1} p_j = 1.$ Обобщенная форма таблицы «объект-свойство», содержащей исходные данные для построения дерева решений приведена в таблице 7;

***Таблица 7. Таблица «объект-свойство» с исходными данными для построения дерева решений***

| | | Предикторы ($W$) | | | | Целевой атрибут ($V$ ,Target) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $W^1$ | $W^2$ | | $W^{n2}$ | $v_1$ | $v_2$ | … | $v_{n1}$ |
| $A$ | $a_1$ | $r_1^1$ | $r_1^2$ | … | $r_1^{n2}$ | *FALSE* | *FALSE* | … | *TRUE* |
| | $a_2$ | $r_2^1$ | $r_2^2$ | … | $r_2^{n2}$ | *FALSE* | *TRUE* | … | *FALSE* |
| | … | … | … | … | … | … | … | … | … |
| | $a_{n4}$ | $r_{n3}^1$ | $r_{n3}^2$ | … | $r_{n3}^{n2}$ | *TRUE* | *FALSE* | … | *FALSE* |

$Q$ – конечное множество из $m$ бинарных тестов или вопросов, $Q_1,Q_2,...,Q_m$, каждый из которых, является функцией $Q_j(a_k)$: $\{a_1,a_2,...,a_{n4}\} \to \{TRUE,FALSE | \forall v_j \in V\}$ , где $Q_j(a_k)$ определяет результат применения теста $Q_j$ для объекта $a_k \in A$. Каждый тест (правило дерева решений) соответствует импликации в синтаксисе логики высказываний: $Q_j : x_1 \wedge x_2 \wedge ... \wedge x_{n_7} \to y$ , где $x_1, x_2, ..., x_{n_7}$ определены на элементах множества $W$ - $\{r_j^i | r_j^i \in W^i\}$ , $n_7$ – количество элементов в логической формуле, а $y$ определено на множестве $\{v_j | v_j \in V\}$;

$C$ – соответствующее множество из $m$ стоимостей $C_1,C_2,...,C_m$ , где $C_j$ затраты на выполнение теста $Q_j$, каждое $C_j > 0$.

Необходимо найти множество тестов суммарной минимальной стоимости, необходимых для однозначной правильной идентификации каждого объекта $a_k \in A$, т.е. таких тестов, которые для объекта $a_k$ принимают значение ИСТИНА только для значения $v_{a_k} \in V$ и значение ЛОЖЬ для всех остальных значений $v_j \in V$.

Табличная форма описания задачи приведена в таблице 8. В столбцах Target приведены значения целевого атрибута, в строках тесты (логические правила), а в ячейках – результат применения теста $Q_j(a_k)$ к объекту $a_k \in A$ для различных значений целевого атрибута.

***Таблица 8. Табличная форма описания задачи построения оптимального дерева решений***

| | | Target | | | |
|---|---|---|---|---|---|
| | | $p_1$ | $p_2$ | … | $p_{n1}$ |
| | $V$ | $v_1$ | $v_2$ | … | $v_{n1}$ |
| $C_1$ | $Q_1(a_k)$ | *FALSE* | *FALSE* | … | *FALSE* |
| $C_2$ | $Q_2(a_k)$ | *FALSE* | *TRUE* | … | *TRUE* |
| … | … | … | … | … | … |
| $C_m$ | $Q_m(a_k)$ | *TRUE* | *FALSE* | … | *FALSE* |

Если сравнить Таблицы 6 и 8, то очевидно, что постановки задач построения оптимального дерева решений и оптимальной процедуры бинарной идентификации схожи. Однако в задаче построения процедуры бинарной идентификации стоимость теста $C_m$ никак не связана ни со свойствами теста, ни с вероятностями наблюдения объектов. Рассмотрим алгоритм построения оптимального дерева решений на основе анализа таблицы «объект-свойство» (табл. 7) при условии задания стоимости теста как функции свойств теста и вероятности наблюдения значения целевого атрибута.

## Процедура построения оптимального дерева решений и ее вычислительная сложность

Уточним определение понятия оптимального дерева решений, построенного на основе анализа таблицы «объект-свойство». Оптимальным будем считать дерево, которое содержит минимальное количество правил (имеющих максимальную поддержку), которые однозначно и безошибочно (со 100% достоверностью) определяют значения целевого атрибута для каждого объекта, входящего в таблицу «объект-свойство». Кроме этого, в соответствии с выводами статистической теории обучения распознаванию [Вапник и Червоненкис, 1974], должны быть отобраны более простые правила, т.е. включающие минимальное количество элементов (предикторов) при прочих равных характеристиках (поддержка и достоверность).

Таким образом, процедура построения оптимального дерева решений включает 2 этапа:

1) Нахождение всех возможных правил на основе анализа таблицы «объект-свойство;
2) Отбор наилучших правил из найденных на первом этапе.

Рассмотрим первый этап.

Найдем какое максимальное количество тестов в синтаксисе логики высказываний в ДНФ может быть задано таблицей «объект-свойство». Тестом является каждая ячейка таблицы, соответствующая определенному значению атрибута объекта. Таким образом, множество всех возможных тестов включает множество *W* (таблица 7), а максимальное количество таких тестов равно $|W|$. Каждый объект или строка таблицы также является тестом (соответствует конъюнкции всех атрибутов объекта), т.е. в множество всех возможных тестов входит множество *A* . Количество таких тестов равно $|A|$. Тесты, которые представлены множеством *A*, имеют 100% достоверность, потому что по условию задачи для каждого объекта $a_k \in A$ определено только одно значение целевого атрибута $v_{a_k} \in V$ . Правда такие тесты не решают задачи построения обобщенной логической формулы класса объектов, потому что обладают относительной поддержкой равной $1/|A|$. Какие еще тесты, пригодные для идентификации значений целевого атрибута (target), может задавать таблица объект-свойство? Такими могут быть фрагменты описаний объектов, которые повторяются в описаниях не менее двух различных объектов. Фрагменты описаний объектов, которые не повторяются, не могут правилами (тестами). Выделять такой фрагмент из полного описания объекта в качестве теста, а не использовать в качестве теста полное описание объекта, нет каких-либо оснований. Оценим теоретически возможное количество фрагментов описаний объектов, которые повторяются в описаниях различных объектов из таблицы «объект-свойство» более одного раза. Для нахождения таких фрагментов необходимо выполнить операцию сравнения описаний объектов друг с другом. Формально, для $\forall a_j \in A$ и $\forall a_i \in A, i, j = \overline{1,n_4}, i \neq j, n_4 = |A|$ , необходимо найти $R_{ji} = R_{a_j} \cap R_{a_i}$ и если $R_{ji} \neq \varnothing$ сформировать тест (правило) $Q_{ij} = \bigvee_{l=1}^{n7} r_l \left| r_l \in R_{ji}, n_7 = |R_{ji}| \right.$. Таким образом максимально возможное количество тестов будет равно сумме членов арифметической прогрессии,

от 1 до $|A|-1$ с шагом 1, если пересечения описаний всех объектов различны и ни одно из них не равно пустому множеству. По формуле суммы членов арифметической прогрессии количество таких тестов равно $|A|\cdot(|A|-1)/2$. Тогда максимальное количество тестов $m$, которое задает исходная таблица «объект-свойство», можно найти по формуле: $m \le |A|\cdot(|A|-1)/2 + |W| + |A| = \frac{|A|^2+|A|}{2} + |W|$.

Количество операций для нахождения всех возможных тестов можно оценить как $O\left(\frac{|A|^2-|A|}{2}\right)$.

Рассмотрим второй этап. В качестве критерия, характеризующего качество правила, можно использовать меру информационного выигрыша, выраженную через количество информации. Рассмотрим ее более подробно. Базовым понятием всей теории информации является понятие энтропии. Информационная энтропия – мера неопределённости некоторой некоторой ситуации или системы, в частности мера непредсказуемости появления какого-либо символа первичного алфавита. При отсутствии информационных потерь энтропия численно равна количеству информации на символ передаваемого сообщения. Информационная двоичная энтропия для атрибута (случайной величины) рассчитывается по формуле Шеннона: $H(Y) = -\sum_{i=1}^{n} \frac{N_i}{N} \bullet \log_2\left(\frac{N_i}{N}\right)$, где $n$ — количество значений атрибута, $N_i$ – количество примеров, которые имеют i-ое значения атрибута, $N$ – общее число примеров в множестве. Фактически информационная двоичная энтропия определяет минимальное число бит, которые необходимы для кодирования выбранного атрибута для надежной передачи информации в виде двоичных чисел. Энтропия, в отличии от дисперсии, не зависит от типа распределения вероятностей случайных величин. Если две случайные величины $X$ и $Y$, каким-то образом связаны

друг с другом, то знание одной из них, уменьшает неопределенность значений другой [Коротаев, 2003]. Оставшаяся неопределенность оценивается условной энтропией. Условная энтропия $X$ при условии знания $Y$ определяется как: $H\left(X\middle|Y\right)=\sum_{k=1}^{K}P\left(Y_k\right)\sum_{m=1}^{M}P\left(X_m\middle|Y_k\right)\bullet\log_2\left(P\left(X_m\middle|Y_k\right)\right)$, где – условные вероятности (вероятность $m$-го значения $X$ при условии $Y=Y_k$), количество значений случайных величин $X$ и $Y$ ($M$ и $K$) не обязательно совпадают. Чтобы рассчитать $H\left(X\middle|Y\right)$, рассчитывают $K$ энтропий $X$, соответствующих фиксированному $Y_k$ далее суммируют результаты с весами $P\left(Y_k\right)$. Условная энтропия всегда меньше безусловной, точнее:

$$0\le H\left(X\middle|Y\right)\le H\left(X\right).$$

Нулевое значение условной энтропии соответствует однозначной зависимости $X$ от $Y$, максимальное значение – полной независимости $X$ и $Y$.

Условная энтропия – это предельно общая характеристика степени зависимости некоторых переменных [Коротаев, 2003]. Ее можно сравнить с корреляцией, но если корреляция характеризует линейную связь переменных, то условная энтропия характеризует любую связь. Информационный выигрыш от использования одной случайной величины для прогнозирования другой случайной величины определяется разностью между безусловной и условной энтропиями этих случайных величин.

В задаче построения дерева решений (табличная форма описания задачи приведена в таблице 8) меру информационного выигрыша для каждого правила можно вычислить по формуле:

$Gain\left(Q_j\right)_{v_i}=H\left(target=v_i\right)-H\left(target=v_i\middle|Q_j\right)$, где $Gain\left(Q_j\right)_{v_i}$ – мера информационного выигрыша от использовании правила $Q_j$ для определения значения целевого атрибута $v_i$ на множестве объектов в

исходной таблице «объект-свойство»; $H(target = v_i)$ – информационная энтропия $v_i$ значения целевого атрибута (target), $H(target = v_i | Q_j)$ – условная энтропия $v_i$ значения целевого атрибута при условии использовании правила $Q_j$.

Информационная энтропия и условная энтропия вычисляются на множестве объектов из исходной таблице «объект-свойство». Для упрощения дальнейших рассуждений предположим, что для каждого значения целевого атрибута существует не менее одного теста, условная энтропия которого на множестве объектов из таблицы «объект-свойство» равна 0.

Второй этап алгоритма построения оптимального дерева решений включает следующие шаги.

1) Для каждого теста $Q_i$, $i = \overline{1,m}$ и всех объектов $a_k \in A, k = \overline{1,n_4}, n_4 = |A|$ определить $v_i \in V$ для которых $Q_i(a_k) = True$ и для каждого $v_j \in V, j = \overline{1,n_1}, n_1 = |V|$ сформировать множества объектов $V_{v_j}^{Qi} = \{a_k | Q_i(a_k) = True\}$. На данном шаге выполняется добавление объектов в множества по значениям целевого атрибута, для которых текущий тест принимает принимает значение истина. Определим систему множеств $X^{Qi} = \left\{ V_{v_j}^{Q_i} \middle| V_{v_j}^{Q_i} \neq \varnothing \right\}$. Если $X^{Qi}$ содержит более одного множества то необходимо удалить данный тест из множества $Q$. Таким образом в множестве $Q$ остаются тесты, которые принимают значения истина только для одного значения целевого атрибута на множестве объектов из исходной таблицы «объект-свойство». Количество операций на данном шаге можно оценить как $O\left( m \cdot |A| = \left( \frac{|A|^2 + |A|}{2} + |W| \right) \cdot |A| = \frac{|A|^3 + |A|^2}{2} + |W| \cdot |A| \right)$.

2) Для каждого теста $Q_i$ из множества $Q$ для всех объектов $a_k \in V_{v_j}^{Qi}$ проверить выполнение условия $v_{a_k} = v_j$. Если условие не выполняется, то удалить данный тест из множества $Q$. Количество операций на данном шаге не превышает $O\left(m \cdot |A|\right)$. Шаги 1 и 2 алгоритма решают задачу отбора правил со 100% достоверностью.

3) Определить значение стоимости $C_i$ теста $Q_i$ в соответствии со свойствами теста и требованиями, которые предъявляются к наилучшему тесту:

$$C_j = F\left(Gain\left(Q_j\right)_{v_i}, p_j, n7\right)$$

Мера информационного выигрыша $Gain\left(Q_j\right)_{v_i}$ никак не учитывает количество элементов в логической формуле теста $Q_i$. Тесты, состоящие из минимального числа элементов, являются лучшими при прочих равных характеристиках [Вапник и Червоненкис, 1974]. Поэтому в функцию стоимости теста необходимо в качестве аргумента добавить $n_7$ – количество элементов в логической формуле. Для минимизации количества вычисляемых тестов, вначале должны вычисляться наиболее вероятные тесты, т.е. такие, которые проверяют наличие у объекта наиболее вероятного значения целевого атрибута – $p_j$. Лучшие тесты должны иметь минимальную стоимость, поэтому стоимость теста должна быть прямо пропорционально количеству элементов в логической формуле, обратно пропорционально вероятности значения целевого атрибута и мере информационного выигрыша. В качестве функции $F$, удовлетворяющей приведенным требованиям может быть выбрана, например, следующая:

$$C_j = \left(1 \Big/ Gain\left(Q_j\right)_{v_i}\right).\left(1 / p_j\right) \cdot n7$$

Количество операций на данном шаге не превышает $m$.

4) Отсортировать множество тестов $Q$ по возрастанию $C_i$. Количество операций на данном шаге не превышает $O\left(|m|\cdot\log\left(|m|\right)\right)$.

5) Для каждого объекта $a_k \in A$ последовательно проверить истинность тестов из отсортированного множества $Q$ в порядке возрастания стоимости теста – $C$. При получении результата истина, отметить данный тест и перейти к следующему объекту $a_k \in A$. Количество операций на данном шаге не превышает $m\cdot|A|$. Использование меры информационного выигрыша на 3 шаге и шаги 4 и 5 предназначено для отбора правил с максимальной поддержкой. Шаг 5 предназначен для выполнения условия определения значения целевого атрибута для каждого объекта, входящего в таблицу «объект-свойство», с помощью выбранных тестов.

6) Отмеченные тесты из $Q$ представляют собой множество тестов минимальной стоимости, необходимых для однозначной идентификации каждого объекта из множества $A$ и соответствуют оптимальному дереву решений в форме множества логических функций.

Каждый шаг приведенного алгоритма имеет полиномиальную сложность (сложность ограничена полиномом третьей степени мощности множества объектов обучающей выборки). Таким образом сложность приведенного алгоритма полиноминальна.

## Выводы и дискуссия

В работе приведена формальная постановка задачи построения оптимального дерева решений на основе анализа таблицы «объект-свойство» в терминах задачи бинарной идентификации. Рассматривается случай, когда все атрибуты имеют конечное число дискретных значений. В работе показано, что существует полиномиальный алгоритм построения оптимального дерева решений при условии определения стоимости теста как функции свойств теста (логического правила). Задача построения оптимального дерева решений не является NP-полной задачей при

условии задания ограничений на определение функции стоимости логического правила. Можно показать, что принятое допущение о существовании для каждого значения целевого атрибута не менее одного теста с условной энтропией равной 0 на множестве объектов из таблицы «объект-свойство» не влияет на полученный вывод о полиномиальной сложности задачи построения оптимального дерева решений. Для построения полиномиального алгоритма решения задачи в этом случае необходимо предварительно построить специальную структуру, в которой логические правила (тесты) упорядочиваются по вхождению более короткого правила в более длинное и добавить операцию отрицания в логические правила. Примером подобной структуры может служить растущая пирамидальная сеть, в которой определен алгоритм нахождения логических правил [Гладун, 1994]. Детальное описание алгоритма построения оптимального дерева решений на основе растущей пирамидальной сети и оценка его сложности будет приведено в дальнейших работах.

**Литература**

[Субботин, 2019] Субботин С. А. Построение деревьев решений для случая малоинформативных признаков // Радіоелектроніка, інформатика, управління. e-ISSN 1607-3274. – 2019. – № 1. – с.122-131. DOI 10.15588/1607-3274-2019-1-12

[Quinlan, 1986] Quinlan J. R. Induction of decision trees // Machine learning. – 1986. – V. 1, № 1. – pp. 81–106.

[Xindong at all, 2008] Wu Xindong, Kumar Vipin; J. Ross Quinlan, Ghosh Joydeep, Yang Qiang, Motoda, Hiroshi; McLachlan, Geoffrey J.; Ng, Angus; Liu, Bing; Yu, Philip S.; Zhou, Zhi-Hua (2008-01-01). Top 10 algorithms in data mining. Knowledge and Information Systems. 14 (1): 1–37. doi:10.1007/s10115-007-0114-2. ISSN 0219-3116

[Deductor, 2020] ООО «Лаборатория баз данных» Электронный ресурс, режим доступа: http://www.basegroup.ru.

[Ho,1998] Ho T.K. (1998). The Random Subspace Method for Constructing Decision Forests // IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601.

[Hyafil and Rivest, 1976] Laurent Hyafil, Ronald L. Rivest. Constructing optimal binary decision trees is np-complete // Information Processing Letters. – 1976. –V. 5, № 1. – pp. 15–17. DOI:10.1016/0020-0190(76)90095-8

[Garey, 1972] M. R. Garey. Optimal Binary Identification Procedure // SIAM Journal on Applied Mathematics, Vol. 23, No. 2 (Sep., 1972), pp. 173-186.

[Вапник и Червоненкис, 1974] Вапник В.Н., Червоненкис А.Я. Теория распознавания образов, статистические проблемы обучения. – М.: Наука, 1974. – 416 с.

[Коротаев, 2003] С. М. Коротаев. Энтропия и информация – универсальные естественнонаучные понятия // Электронный ресурс, режим доступа: http: //www.chronos.msu.ru/old/rreports/korotaev_entropia/korotaev_entropia.htm

[Гладун, 1994] Гладун В.П. Процессы формирования новых знаний. – Sofia: SD “Педагог-6”, 1994. – 192 с.

**Об авторе**

***Виталий Величко*** *– Институт кибернетики им. В. М. Глушкова НАН Украины; Кандидат технических наук. Старший научный сотрудник. Проспект Глушкова 40, Киев, Украина, 03187; e-mail: aduisukr@gmail.com*

*Основные направления научных исследований: индуктивный логический вывод, компьютерные онтологии, информационные системы с обработкой объектов естественного языка*

# To the Problem of Constructing the Optimal Decision Tree

**Vitalii Velychko**

***Abstract****: The paper presents a formal statement of the problem of constructing an optimal decision tree in terms of a binary identification problem. An optimal decision tree is defined as a minimum size tree and capable of classifying all objects from the training set without errors. The case is considered when all attributes of objects are nominal. To select the best rules, a measure of information gain is used based on the calculation of conditional entropy. The paper shows that there is a polynomial algorithm for solving the problem of constructing an optimal decision tree, formulated in terms of a binary identification problem. The condition for the existence of the algorithm is the determination of the cost of the test (logical rule) as a function of the test properties. The computational complexity of the above algorithm is limited by the third degree polynomial of the power of the set of objects of the training sample. For the sake of simplicity, it is assumed that for each value of the target property there is at least one test, the conditional entropy of which on the set of objects from the training set is equal to 0. The problem of constructing an optimal decision tree is not an NP-complete problem if constraints are set on determining the cost function of a logical rule (test).*

***Key words:*** *binary identification problem, optimal decision tree, computational complexity class P.*