$$\text{MIMPS} = \text{Imps}\left(U_i, A_j, T_k, T_{k+1}, \left\{\hat{\mathbf{U}}\right\}\right), \text{IMPS} = \text{Imps}\left(A_j, T'_k, T'_{k+1}, \left\{\hat{\mathbf{U}}\right\}\right),$$

где

$$\text{MDSTN} = \begin{cases} 1, \text{Imps}\left(U_i, A_j, T_k, T_{k+1}, \left\{\hat{\mathbf{U}}\right\}\right) \neq 0 \\ 0, \text{Imps}\left(U_i, A_j, T_k, T_{k+1}, \left\{\hat{\mathbf{U}}\right\}\right) = 0 \end{cases}, \text{DSTN} = \text{Distinct}\left(A_j, T'_k, T'_{k+1}, \left\{\hat{\mathbf{U}}\right\}\right).$$

Выводы

Проведена разработка алгоритма прогнозирования взаимодействий пользователей с рекламными элементами интернет-страниц, формализована математическая модель.

В качестве дальнейшей работы автором будет производена детальная оценка качества предложенного алгоритма при различных случаях распределения значений временных рядов, производится улучшение метода построения одномерного прогноза, скорость его выполнения и его точность.

М.А. Борисов, Н.Г. Крупец

АЛГОРИТМЫ ПОСТРОЕНИЯ ДЕРЕВА ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ КЛАССИФИКАЦИИ

(Самарский университет)

В представленной работе рассматриваются методы построения правил классификации объектов, характеризуемых вектором признаков, измеренных в различных шкалах измерения.

Целью работы является исследование точностных характеристик алгоритмов составления дерева принятия решений, позволяющего решать задачи классификации на моделях кластеров обучающих и тестирующих последовательностей объектов.

Дерево принятия решений (также может называться деревом классификации или регрессионным деревом) - средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике [1]. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны атрибуты, от которых зависит целевая функция (функция классификации объектов), в «листьях» записаны значения целевой функции, в остальных узлах — атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение.

Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе.

Каждый лист представляет собой значение целевой переменной, измененной в ходе движения от корня по листу. Каждый внутренний узел соответствует одной из входных переменных.

Дерево может быть также «изучено» разделением исходных наборов переменных на подмножества, основанные на тестировании значений атрибутов. Этот процесс повторяется на каждом из полученных подмножеств. Рекурсия завершается тогда, когда подмножество в узле имеет те же значения целевой переменной, т.е. оно не добавляет ценности для предсказаний.

В представленной работе рассмотрено 3 алгоритма построения дерева; ID3, C4.5, CART.

Алгоритм ID3 — один из алгоритмов для построения дерева принятия решений, разработанный Джоном Р. Квинланом [2]. Алгоритмы ID3 предназначен для работы с атрибутами (признаками), измеренными в шкале наименований и порядка (с нечисловыми атрибутами).

Описание алгоритма ID3

Пусть мы имеем проверку X (в качестве проверки может быть выбран любой атрибут), которая принимает n значений A_1, A_2 ... A_n. Тогда разбиение T по проверке X даст нам подмножества T_1, T_2 ... T_n, при X равном соответственно A_1, A_2… A_n. Единственная доступная информация – это, каким образом классы распределены в множестве T и его подмножествах, получаемых при разбиении по X. Таким образом определение критерия для выбора атрибута будет следующим.

Пусть $freq(C_j, S)$– количество примеров из некоторого множества S, относящихся к одному и тому же классуC_j.

Согласно теории информации, количество содержащейся в сообщении информации, зависит от ее вероятности:

$$\log_2\left(\frac{1}{P}\right) \quad (1)$$

Поскольку мы используем логарифм с двоичным основанием, то выражение (1) дает количественную оценку содержащейся информации в битах.

$$Info(T) = -\sum_{j=1}^{k} \frac{freq(C_j,T)}{|T|} * \log_2 \frac{freq(C_j,T)}{|T|} \quad (2)$$

Выражение (2) дает оценку среднего количества информации, необходимого для определения класса примера из множества T. В терминологии теории информации выражение (2) называется энтропией множества T.

Ту же оценку, но только уже после разбиения множества T по X, дает следующее выражение:

$$Info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} * Info(T_i) \quad (3)$$

Тогда критерием для выбора атрибута будет являться следующая формула:

$$Gain(X) = Info(T) - Info_x(T) \quad (4)$$

Критерий (4) считается для всех атрибутов. Выбирается атрибут, максимизирующий данное выражение. Этот атрибут будет являться проверкой в текущем узле дерева, а затем по этому атрибуту производится дальнейшее по-

строение дерева. Т.е. в узле будет проверяться значение по этому атрибуту, и дальнейшее движение по дереву будет производиться в зависимости от полученного ответа.

Впоследствии автором была создана усовершенствованная версия алгоритма ID3 - алгоритм C4.5 [3]. В алгоритм C4.5 в сравнении с ID3 были добавлены отсечение ветвей (англ. pruning), возможность работы с числовыми атрибутами, а также возможность построения дерева из неполной обучающей выборки, в которой отсутствуют значения некоторых атрибутов.

Алгоритм разбиения по числовому атрибуту C4.5.

Следует выбрать некий порог, с которым должны сравниваться все значения атрибута. Пусть числовой атрибут имеет конечное число значений. Обозначим их $\{v_1, v_2 \dots v_n\}$. Предварительно отсортируем все значения. Тогда любое значение, лежащее между v_i и v_{i+1}, делит все примеры на два множества: те, которые лежат слева от этого значения $\{v_1, v_2 \dots v_i\}$, и те, что справа $\{v_{i+1}, v_{i+2} \dots v_n\}$. В качестве порога можно выбрать среднее между значениями v_i и v_{i+1}:

$$TH_i = \frac{v_i + v_{i+1}}{2} \quad (5)$$

Таким образом, существенно упрощается задача нахождения порога, и приводятся к рассмотрению всего n-1 потенциальных пороговых значений $TH_1, TH_2, \dots TH_{n-1}$.

Формулы (2), (3) и (4) последовательно применяются ко всем потенциальным пороговым значениям, и среди них выбирается то, которое дает максимальное значение по критерию (4). Далее это значение сравнивается со значениями критерия (4), подсчитанными для остальных атрибутов. Если выяснится, что среди всех атрибутов данный числовой атрибут имеет максимальное значение по критерию (4), то в качестве проверки выбирается именно он.

Следует отметить, что все числовые тесты являются бинарными, т.е. делят узел дерева на две ветви [4].

Алгоритм CART. Алгоритм предназначен для построения бинарного дерева решений. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров на две части — часть, в которой выполняется правило (потомок — right) и часть, в которой правило не выполняется (потомок — left) [5].

Преимуществом алгоритма CART является определенная гарантия того, что если искомые детерминации существуют в исследуемой совокупности, то они будут выявлены. Кроме того, CART позволяет не «замыкаться» на единственном значении выходного признака, а искать все такие его значения, для которых можно найти соответствующее объясняющее выражение.

В алгоритме CART идея неопределенности формализована в индексе *Gini*. Если набор данныхT содержит данные n классов, тогда индекс *Gini* определяется следующим образом:

$$Gini(T) = 1 - \sum_{i=1}^{n} p_i^2 \quad (6)$$

где p_i — вероятность (относительная частота) класса *i* в *T*. Если набор *T* разбивается на две части *T1* и *T2* с числом примеров в каждом *N1* и *N2* соответственно, тогда показатель качества разбиения будет равен:

$$Gini_{split}(T) = \frac{N_1}{N} * Gini(T_1) + \frac{N_2}{N} * Gini(T_2) \quad (7)$$

Наилучшим считается то разбиение, для которого $Gini_{split}(T)$ минимально.

Данные алгоритмы были реализованы и протестированы в виде программ на языке Java в API Weka. В качестве обучающих последовательностей были использованы модели кластеров с нормальным законом распределения признаков, измеренных в шкале отношений в n-мерном пространстве.

Представленные алгоритмы тестировались на моделях кластеров, сгенерированных следующим образом. Первый кластер моделировался последовательностью нормально распределенных векторов с заданным вектором математических ожиданий и вектором дисперсий. Второй кластер формировался аналогичным образом с вектором математических ожиданий, отстоящих от вектора математических ожиданий первого класса на расстояние R и вектором дисперсий в δ раз больше. Далее значения признаков, смоделированных как измеренных в шкале отношений, преобразовывались в значения признаков, измеренных в шкале наименований или порядка, методом «попадания в пронумерованную прямоугольную коробку».

Для оценки точности построенного решающего правила была использована оценка вероятности правильной классификации суммарно по двум классам смоделированной тестовой последовательности объектов.

Результаты показали, что при длине обучающей выборки 300-500 экземпляров и размерности пространства 10-20 признаков, наибольший эффект по оценке вероятности правильной классификации на тестируемой последовательности объектов является алгоритм C4.5.

Литература

1 Дерево решений [Электронный ресурс] /. https://ru.wikipedia.org/wiki/Дерево_решений

2 ID3 [Электронный ресурс] /. https://ru.wikipedia.org/wiki/ID3_(алгоритм)

3 Quinlan *J. R.* Learning With Continuous Classes (англ.) // Proceedings of the 5th Australian Joint Conference on Artificial Intelligence. — 1992. — P. 343—348. — ISBN 98-9810-2125-06.

4 Quinlan *J. R.* C4.5: Programs for Machine Learning. — San Mateo: Morgan Kaufmann Publishers Inc., 1993. — 302 p. — ISBN 1-5586-0238-0. (англ.)

5 Деревья решений - CART математический аппарат. [Электронный ресурс] /. http://www.basegroup.ru/trees/math_cart_part1.html