

УДК 519.714

СРАВНЕНИЕ АЛГОРИТМОВ РАСШИФРОВКИ МОНОТОННЫХ ФУНКЦИЙ МЕТОДОМ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ¹⁾

© 1999 г. Д. Л. Кожевников, О. И. Ларичев

(117312 Москва, пр-т 60-летия Октября, 9, Ин-тут системного анализа РАН)

Поступила в редакцию 20.05.98 г.

Алгоритмы расшифровки монотонных функций находят практическое применение в задачах построения баз экспертных знаний. В этих задачах важны не только оценки оптимальности алгоритма в наиболее сложном случае, но и оценки его эффективности при произвольных границах между классами решений. В статье получены оценки качества четырех алгоритмов методом статистического моделирования.

1. ЗАДАЧА РАСШИФРОВКИ МОНОТОННЫХ ФУНКЦИЙ

Задача расшифровки монотонных функций алгебры логики может быть кратко представлена следующим образом [1]. Рассмотрим класс функций, определенных на конечном множестве и принимающих значения из множества $\{0, 1\}$. Функция f называется монотонной, если из условия $b_i \leq b_j$ следует, что

$$f(b_i) \leq f(b_j).$$

Пусть имеется источник информации, который назовем *оракулом*. Под оракулом можно понимать как объективный источник информации (например, измерительное устройство), так и человека (лицо, принимающее решения, эксперт).

Класс всех монотонных булевых функций от N переменных обозначим через M_N . Любая функция $f \in M_N$ задает разбиение множества E^N на два непересекающихся подмножества. Обозначим их через f_1 (множество нулей) и через f_2 (множество единиц).

Определение 1. Максимальные элементы множества f_1 и минимальные элементы множества f_2 называются граничными элементами.

Определение 2. Назовем множества f_1 и f_2 классами решений.

Задача расшифровки монотонных функций с оракулом состоит в следующем (см. [1]). Предполагается, что оракул дает безошибочный ответ на вопрос, каково значение функции f в данной точке $\alpha \in E^N$. Иначе говоря, оракул определяет, к какому классу, f_1 или f_2 , следует отнести данную точку. Вопросы к оракулу ставятся до тех пор, пока по его ответам либо из свойства монотонности будут определены значения функции f во всех точках E^N .

Одной из известных задач алгебры логики является задача разработки алгоритмов оптимальной расшифровки монотонных функций [1]–[3]. Созданы эффективные алгоритмы расшифровки, получены оценки числа обращений к оракулу [1], [3]. Минимаксная оценка числа обращений определяется, как функция Шеннона [4], в виде

$$\varphi(N) = C_N^{N/2} + C_N^{N/2+1}.$$

Наряду с теоретической важностью алгоритмы расшифровки монотонных функций находят практическое применение в задачах искусственного интеллекта.

2. ЗАДАЧА ЭКСПЕРТНОЙ КЛАССИФИКАЦИИ

Близкой по постановке к описанной выше задаче является одна из задач искусственного интеллекта – задача экспертной классификации [5]. Эта задача может быть представлена в следующем виде.

¹⁾ Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (коды проектов 98-01-00086 и 96-15-96155).

Дано: N – число признаков, описывающих рассматриваемые объекты, w_i – шкала дискретных значений i -го признака, для каждого класса они могут быть упорядочены от более к менее характерным; n_i – количество значений на шкале i -го признака; $X_i = \{x_{ij}\}$ – множество всех возможных значений по i -му критерию, $j = 1, 2, \dots, n_i$; $|X_i| = n_i$; $Y = X_1 \cdot X_2 \cdot \dots \cdot X_N$ – множество векторных оценок вида $y_j = (y_{j1}, \dots, y_{jN})$, где $y_{jp} \in X_p$; K – число классов решений.

Требуется: на основе информации эксперта построить классификацию всех векторов y_j , отнеся их к классам решений.

В [5], [6] даны различные алгоритмы решения этой задачи.

Отличие задачи расшифровки монотонных функций от задачи экспертной классификации заключается в следующем.

1. Задача экспертной классификации имеет более общий характер: несколько классов решений; могут быть пересекающиеся классы; немонотонные функции; произвольное число оценок на шкалах признаков; разное упорядочение оценок для различных классов.

2. Эксперт и оракул – не одно и то же. Эксперт может делать ошибки. Поэтому в алгоритмах решения задач экспертной классификации [5]–[7] предусмотрены процедуры поиска и устранения ошибок.

В методах решения задач экспертной классификации одним из важных является этап определения наиболее эффективной стратегии предъявления векторов y_i эксперту для их классификации. Этот этап направлен на решение той же задачи, что и алгоритмы расшифровки монотонных функций. Если взять $n_i = 2$, $K = 2$ и предположить, что функция монотонна и эксперт не ошибается, то обе задачи совпадают. Критерий построения непротиворечивой классификации за минимальное число обращений к эксперту вполне естествен в задачах экспертной классификации. Для таких задач разработаны различные алгоритмы построения баз знаний, имитирующих суждения эксперта [5]–[7]. Однако систематическое сопоставление этих алгоритмов, а также алгоритмов расшифровки монотонных функций ранее не проводилось.

3. ПОСТАНОВКА ЗАДАЧИ

Целью данной статьи является изложение результатов сравнения различных алгоритмов расшифровки монотонных функций применительно к задаче экспертной классификации. Слово “применительно” означает, что критерии оценки алгоритмов и подход к сравнению отражают специфику задач построения баз экспертных знаний. В задачах экспертной классификации нас интересует оценка качества алгоритма при различных типах границ между классами решений. Функция, определяющая границу между классами, отражает знания эксперта. Следовательно, алгоритм должен быть эффективен при любой функции определенного вида (см. ниже).

Одним из основных требований, предъявляемых к алгоритму, является его “приспособляемость” к различным функциям. Иначе говоря, хотелось бы, чтобы алгоритм был близок к оптимальному при различных функциях.

Нахождение аналитических оценок качества алгоритма для различного вида функций (границ между классами решений) – очень сложная задача. Поэтому целесообразно осуществить сравнение различных алгоритмов методом статистического моделирования.

Общая схема сравнения может быть представлена следующим образом:

- 1) моделируются функции, разделяющие классы решений, которые задают ответы “оракула”;
- 2) различные алгоритмы классификации (расшифровки функций) восстанавливают на основе ответов оракула конкретную функцию;
- 3) качество каждого из алгоритмов оценивается по определенным критериям, причем подсчитывается среднее значение по совокупности испытаний.

4. РЕШАЮЩИЕ ПРАВИЛА ЭКСПЕРТОВ

Психологические исследования показывают, что решающие правила экспертов, определяющие границы между классами решений, имеют вполне определенную структуру [8]. Внутренняя организация экспертных знаний определяется характеристиками человеческой системы переработки информации. За время многолетней практики эксперт (врач, геолог, инженер) вырабатывает правила (подсознательные и невербализуемые) отнесения объектов с определенными характеристиками к классам решений. Как показывают исследования (см. [8], [9]), число таких

правил невелико. Каждое из правил в общем случае имеет структуру дерева, в корне которого находятся сочетания значений t наиболее важных, основных признаков. К ним добавляется некоторое количество менее важных признаков, характерных для данного класса. Обычно мало-важные признаки "взаимозаменяемы" и правило их добавления к корню дерева имеет вид сочетания S_p^k , где p – общее число мало-важных признаков ($N = p + t$), k – число таких признаков, которое следует присоединить к основным для принятия определенного решения.

Следовательно, при моделировании необходимо выбирать расшифровываемые функции, имеющие описанную выше структуру.

5. КРИТЕРИАЛЬНЫЕ ОЦЕНКИ АЛГОРИТМОВ

С практической точки зрения логично оценивать две основные характеристики каждого алгоритма. Прежде всего, если задан определенный вид функций (число основных признаков, правило принятия решений), то алгоритм характеризуется средним числом вопросов к оракулу, необходимым для расшифровки функции.

Легко увидеть, что граничные объекты полностью характеризуют расшифровываемую функцию. В связи с этим в качестве второго критерия логично выбрать отношение среднего числа вопросов к числу граничных объектов.

В качестве третьего критерия возможно выбрать количество вычислений, необходимое на каждом шаге алгоритма. Однако его можно использовать как ограничение. Мы исключаем из рассмотрения переборные алгоритмы. В связи с этим не рассматривается один из первых алгоритмов экспертной классификации – поиск информативных точек [5].

6. СРАВНИВАЕМЫЕ АЛГОРИТМЫ

Для сравнения различных алгоритмов будем рассматривать признаки (критерии) с бинарными шкалами и два класса решений. Предполагается также, что рассматриваемая функция монотонна.

Рассмотрим следующие алгоритмы.

1. *Разбиение на цепи* (см. [10]). Основная идея данного алгоритма заключается в сведении исходной задачи к набору подзадач. Исходное множество векторных оценок разбивается на непесекающиеся и линейно упорядоченные множества – цепи. Решается задача порядковой классификации на каждом из полученных подмножеств.

Определение 3. Последовательность наборов (векторных оценок) $u_1, \dots, u_s$ из Y такая, что $u_1 P^1 u_2 P^1 \dots u_s$, называется цепью (P^1 – отношение доминирования). Число наборов в цепи называется длиной цепи. Отдельный набор является цепью длины 1.

Будем говорить, что выбор u непосредственно следует за набором x , если он получается из x увеличением на 1 ровно одной координаты.

Про три набора x, y, z , непосредственно следующих один за другим, скажем, что они коллинеарные, если y получается из x и z получается из y увеличением на 1 одной и той же координаты. В противном случае скажем, что они не коллинеарны.

Определение 4. Пусть наборы x, y, z непосредственно следующие один за другим, не коллинеарны, и пусть y получается из x увеличением i -й координаты на 1, а z получается из y увеличением j -й координаты на 1, $i \neq j$. Тогда набор g , получающийся из x увеличением j -й координаты на 1, назовем сопряженным к тройке векторов x, y, z . Заметим, что при этом набор g получается из y увеличением i -й координаты на 1.

Ключевым результатом, на который опирается алгоритм, является следующая

Лемма 1. *Наборы (векторы) из Y можно разбить на цепи так, что будут выполняться следующие условия:*

- 1) *цепи не пересекаются и содержат все наборы из Y ;*
- 2) *наборы в каждой цепи непосредственно следуют один за другим;*
- 3) *все цепи симметричны относительно среднего слоя, элементы которого имеют половину оценок, принадлежащих каждому из классов;*
- 4) *если x, y, z – любые три не коллинеарных последовательных набора одной цепи длины s , то сопряженный им набор лежит на некоторой цепи длины $s - 2$.*

Доказательство можно найти в [10].

Предположим, что у нас два класса решений. Разобьем все наборы из Y на цепи согласно лемме. Упорядочим полученные цепи по неубыванию длины. Просматривая затем цепи по порядку, будем предъявлять оракулу еще не классифицированные векторы. Пусть C – очередная цепь. Цепь C делится на три участка: сначала идут наборы, которые принадлежат к первому классу, затем наборы, классификация которых еще не определена, и затем наборы, принадлежащие второму классу. Пусть длина участка цепи C , где находятся еще не классифицированные элементы, равна s . Разделим этот участок пополам и предъявим оракулу средний набор (если s нечетно) или один из двух средних (если s четно). Так как функция монотонна, то при этом определяется класс на одном из двух участков. Участок цепи C , оставшийся неклассифицированным, снова делим пополам, и т.д. Нетрудно подсчитать, что для полной классификации на цепи C потребуется запросить ее значения на $\lceil \log_2 s \rceil + 1$ наборах. После классификации всех векторов цепи C переходим к следующей цепи, и т.д.

В [10] дана оценка качества алгоритма. Обозначим через $t(A)$ число предъявлений (обращений к оракулу) алгоритма, необходимое для построения полной классификации.

Сложность алгоритма определяется по формуле $t(A) = \max t_A(f)$, где максимум берется по всем функциям рассматриваемого класса.

Пусть $u_{\text{опт}}$ – любой алгоритм наименьшей сложности. Число предъявлений предложенного алгоритма по отношению к оптимальному равно $t(A) = t(u_{\text{опт}})$.

2. *Пространственная дихотомия* (см. [3]). Предлагаемый алгоритм также использует разбиение задачи на подзадачи. Однако в основе подобного разбиения на этот раз лежит схема пространственной дихотомии. Алгоритм предназначен для разбиения исходного множества Y на два класса.

При четном N единичный N -мерный куб E^N разбивается на четыре подмножества: $E^{N-2}(0, 0)$, $E^{N-2}(0, 1)$, $E^{N-2}(1, 0)$, $E^{N-2}(1, 1)$. Оракулу (эксперту) предъявляются наборы сначала “в середине” единичного N -мерного куба E^N , отдельно в точках множества $E^{N-2}(1, 0)$ и множества $E^{N-2}(0, 1)$, а затем “на концах”, отдельно в точках множества $E^{N-2}(0, 0)$ и множества $E^{N-2}(1, 1)$. При этом каждое из четырех указанных подмножеств разбивается, в свою очередь, на четыре подмножества E^{N-4} , и т.д.

При нечетном N множество E разбивается на два подмножества, $E^{N-1}(0)$ и $E^{N-1}(1)$, для которых задача расшифровки решается отдельно по приведенной выше схеме для четного N . В [3] доказано, что данный алгоритм оптимален по Шеннону.

3. *Дифференциальная классификация* (см. [6]).

Определение 5. Слоем S_j назовем подмножество исходного множества векторных оценок Y , составленное из всех векторов, имеющих среди своих компонент j единиц.

Определение 6. Средним слоем S назовем слой S_m множества Y , векторы в котором имеют равное число нулей и единиц. При нечетном N имеем два средних слоя.

Алгоритм предназначен для разбиения исходного множества Y на два класса.

Пусть средний вектор $u_{\text{ср}}$ принадлежит к первому классу. По определению, это значит, что половина компонент вектора имеет оценки 1, а половина – оценки 0. Граничный элемент расположен в N -мерном пространстве где-то между $u_{\text{ср}}$ и вектором $Y = (1, \dots, 1)$. Объясним все компоненты $u_{\text{ср}}$, имеющие оценки 1, неизменными и будем производить операции замены оценок только для оставшихся оценок.

Разделим изменяемые оценки $u_{\text{ср}}$ пополам при четном $N/2$ (либо на $(N-1)/4$ и $(N+1)/4$ при нечетном). При делении отделяем первую половину изменяемых оценок от остальных. Образует два вектора u_1 и u_2 , имеющих, соответственно, по половине разных компонент с нулевыми оценками из изменяемых оценок $u_{\text{ср}}$. Далее векторы u_1 и u_2 предъявляются оракулу, если их принадлежность к классу решений еще неизвестна либо их принадлежность к определенному классу подтверждается без предъявления. Если какой-то из этих векторов принадлежит снова первому классу, то, аналогично предыдущему, его опять делим на два, считая теперь изменяемой частью компоненты, имевшие нулевые оценки в данном векторе.

Для алгоритма справедлива следующая оценка (см. [6]): отношение числа предъявлений предложенного алгоритма t_1 к оптимальному в случае, если граница проходит по среднему слою, не превышает $t_1/t_{\text{опт}} \leq (l+1)/2$, где $l = P\{\log_2(N/2)\}$ и $P\{q\} \geq q$ – ближайшее к q целое число.

4. *Случайный выбор.* Из еще не классифицированных векторов случайным образом (распределение предполагается равномерным) выбирается вектор для предъявления эксперту. После классификации данного вектора, как и во всех предыдущих алгоритмах, осуществляется косвен-

ная классификация доминируемых и доминирующих ситуаций. Алгоритм останавливается при построении полной классификации.

Этот алгоритм является самым простым из рассматриваемых. Основанием для его применения послужил тот факт, что отсутствует оценка эффективности алгоритмов в общем случае, поскольку наихудший случай для каждого алгоритма встречается редко. Поэтому алгоритм явился некоторым ориентиром при оценке эффективности.

Алгоритм чрезвычайно прост. Какие-либо теоретические оценки его эффективности отсутствуют.

7. СХЕМА ЭКСПЕРИМЕНТА

Статистическое моделирование осуществляется по следующей схеме: строится расшифровываемое изображение, характерное по своему типу для решающих правил экспертов (см. выше), а затем оно расшифровывается приведенными выше алгоритмами.

Расшифровываемое изображение, имеющее структуру двухуровневого дерева, строится следующим образом.

1. Задается количество основных критериев.

2. Случайным образом из еще не классифицированных сочетаний оценок основных критериев выбирается одно сочетание. Затем случайным образом выбирается минимальное количество нулевых оценок по вспомогательным критериям, необходимое для отнесения вектора к первому классу. Конечно, выбор числа вспомогательных критериев с нулевыми оценками осуществляется не совсем случайно, так как компьютер проверяет согласованность, непротиворечивость по отношению к уже классифицированным векторам.

3. Происходит классификация всех оценок с таким сочетанием оценок по основным критериям, а также доминируемых ими и доминирующих их векторов. Подобные действия выполняются до тех пор, пока не будет построена полная классификация.

4. Применяя алгоритм, использующий разбиение на цепи, расшифровываем построенное отображение. После этого расшифровка выполняется алгоритмами случайного выбора ситуации, пространственной дихотомии и дифференциальной классификации.

5. Описанная выше процедура (пп. 1–4) повторяется заданное число раз.

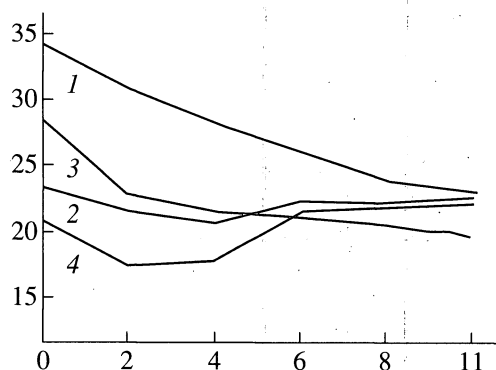
6. Осуществляется оценка алгоритмов по критериям (отношения среднего числа вопросов к числу граничных элементов и среднее число вопросов для каждого из алгоритмов) и вывод результатов.

Эксперимент был проведен для задачи порядковой классификации с 11 двоичными критериями. Общее число векторных оценок составляло 2048 ситуаций. Каждая серия эксперимента состояла из 50 испытаний. Было проведено по 10 серий испытаний для следующих классов отображений.

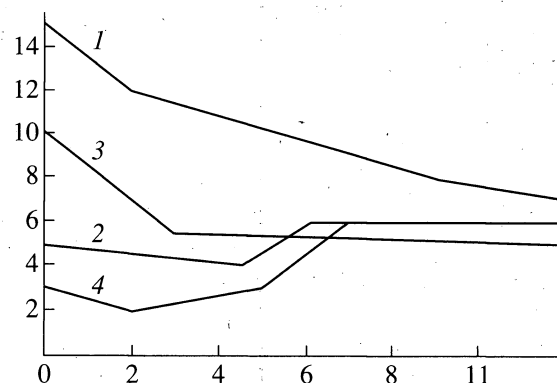
I. Основные критерии отсутствуют. (Имеется всего один критерий с 22 оценками.)

II. 2 основных критерия.

III. 4 основных критерия.



Фиг. 1.



Фиг. 2.

IV. 6 основных критериев.

V. 8 основных критериев.

VI. Все критерии основные. Структура дерева отсутствует.

Таким образом, всего было проведено 3000 испытаний.

8. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Представим результаты в виде таблиц.

Таблица 1

Алгоритмы	Количество основных критериев					
	0	2	4	6	8	11
1. Разбиение на цепи	34.864	30.541	28.131	25.748	24.61	23.513
2. Случайный выбор	24.442	21.544	20.672	22.646	22.219	22.641
3. Пространственная дихотомия	27.47	23.012	21.471	20.881	20.738	20.2
4. Дифференциальная классификация	20.561	18.037	18.788	21.41	21.583	22.472

Таблица 2

Алгоритмы	Количество основных критериев					
	0	2	4	6	8	11
1. Разбиение на цепи	11.483	8.318	5.577	3.325	3.393	3.068
2. Случайный выбор	2.8599	2.752	2.513	2.5211	2.841	2.888
3. Пространственная дихотомия	7.1312	5.461	3.725	2.523	2.718	2.563
4. Дифференциальная классификация	1.1899	1.505	1.921	2.313	2.727	2.863

На фиг. 1 и 2 по оси абсцисс отложено число основных критериев, а цифры при кривых соответствуют номерам алгоритмов в таблицах. По оси ординат на фиг. 1 отложено отношение числа вопросов к общему числу ситуаций в процентах при полной классификации, а на фиг. 2 – отношение числа вопросов к числу граничных элементов при полной расшифровке.

9. ОБСУЖДЕНИЕ

Результаты статистического моделирования показывают, что при произвольной границе между классами и типовой структуре решающих правил эксперта (см. выше) наиболее эффективным “в среднем” является алгоритм дифференциальной классификации. Конечно, при границе максимальной сложности, проходящей по среднему слою, лучшие (хотя и не намного) результаты дает алгоритм пространственной дихотомии. Также следует отметить интересный факт, что чрезвычайно простой алгоритм случайного выбора показывает неплохие результаты.

Полученные результаты позволяют утверждать, что в общем случае алгоритм дифференциальной классификации имеет хорошую “приспособляемость” к различным типам расшифровываемых функций.

10. ЗАКЛЮЧЕНИЕ

Получение аналитических оценок алгоритмов расшифровки монотонных функций алгебры логики является чрезвычайно сложной математической задачей. Оценка, данная для функции наиболее сложного вида, далеко не все говорит о поведении алгоритма в случае функций произвольного вида. В общем случае подход статистического моделирования позволяет получить полезные результаты.

Следует отметить, что теоретические исследования по проблеме расшифровки монотонных функций алгебры логики имеют важные практические приложения.

Авторы считают своим долгом отметить, что встречи и беседы с Н.А. Соколовым были весьма полезны при выполнении данной работы.

СПИСОК ЛИТЕРАТУРЫ

1. Соколов Н.А. Об оптимальной расшифровке монотонных функций алгебры логики // Ж. вычисл. матем. и матем. физ. 1982. Т. 22. № 2. С. 449–461.
2. Коробков В.К. О монотонных функциях алгебры логики // Пробл. кибернетики. 1965. № 13. С. 5–29.
3. Соколов Н.А. Оптимальная расшифровка монотонных булевых функций // Ж. вычисл. матем. и матем. физ. 1987. Т. 27. № 12. С. 1878–1887.
4. Ансель Ж. О числе монотонных булевых функций n переменных // Кибернетич. сб. М.: Мир, 1968. № 5. С. 53–58.
5. Ларичев О.И., Мошкович Е.М., Мечитов А.И., Фуремс Е.М. Выявление экспертных знаний. М.: Наука, 1989.
6. Ларичев О.И., Болотов А.А. Система ДИФКЛАСС: построение полных и непротиворечивых баз экспертных знаний в задачах дифференциальной классификации // Научно-техн. информация. М.: ВИНТИ, 1996. № 9. С. 9–15.
7. Фуремс Е.М., Гнеденко Л.С. STEPCLASS – система извлечения экспертных знаний и проведения экспертизы для решения диагностических задач // Научн.-техн. информация. М.: ВИНТИ, 1996. № 9. С. 16–20.
8. Ларичев О.И. Структуры экспертных знаний // Психология. ж. 1995. Т. 16. № 3. С. 82–89.
9. Larichev O.I. A study on the internal organization of expert knowledge // Pattern Recognition and Image Analysis. 1995. V. 5. № 1.
10. Алексеев В.Б. Расшифровка некоторых классов монотонных многозначных функций // Ж. вычисл. матем. и матем. физ. 1976. Т. 16. № 1. С. 189–198.