

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 512.55

О СХОДИМОСТИ НОВОГО АЛГОРИТМА ХАРАКТЕРИЗАЦИИ k -ЗНАЧНЫХ ПОРОГОВЫХ ФУНКЦИЙ

А. В. Бурделев

Белорусский государственный университет, г. Минск, Беларусь

Функция k -значной логики $f(x_1, \dots, x_n)$, для которой существует линейная форма $L(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$, $x_i \in \mathbb{Z}_k$, с вещественными коэффициентами и набор вещественных порогов $b_0 < b_1 < \dots < b_k$, такие, что для всех $i \in \{0, \dots, k-1\}$ выполняется условие $f(x_1, \dots, x_n) = i \Leftrightarrow b_i \leq L(x_1, \dots, x_n) < b_{i+1}$, называется пороговой k -значной функцией. Под алгоритмом характеристики пороговой k -значной функции понимается процедура нахождения коэффициентов $a_1, a_2, \dots, a_n$ линейной формы $L(x_1, \dots, x_n)$ и множества порогов $b_0, b_1, \dots, b_{k-1}$. В работе доказывается сходимость алгоритма нахождения коэффициентов линейной формы и порогов (характеристики) k -значных пороговых функций по столбцу значений. Основная идея алгоритма заключается в раздельном последовательном вычислении коэффициентов линейной формы и порогов. В качестве первичной аппроксимации линейной формы используются коэффициенты роста либо коэффициенты возрастания и итеративно осуществляется корректировка линейной формы. После нахождения коэффициентов линейной формы вычисляются разделяющие пороги.

Ключевые слова: алгоритм характеристики, доказательство сходимости, пороговая функция.

DOI 10.17223/20710410/39/10

CONVERGENCE OF AN ITERATIVE ALGORITHM FOR COMPUTING PARAMETERS OF MULTI-VALUED THRESHOLD FUNCTIONS

A. V. Burdelev

*Belarus State University, Minsk, Belarus***E-mail:** aburd2011@mail.ru

A k -valued threshold function is defined as $f(x_1, \dots, x_n) = i \in \{0, 1, \dots, k-1\} \Leftrightarrow b_i \leq L(x_1, \dots, x_n) < b_{i+1}$ where $L(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$ is a linear form in variables $x_1, \dots, x_n$ with the values in $\{0, 1, \dots, k-1\}$ and coefficients $a_1, \dots, a_n$ in $\mathbb{R}$ and $b_0, \dots, b_k$ are some thresholds for L in $\mathbb{R}$, $b_0 < b_1 < \dots < b_k$. A. V. Burdelev and V. G. Nikonov have created and published in J. Computational Nanotechnology (2017, no.1, pp.7–14) an iterative algorithm for computing coefficients $a_1, \dots, a_n$ and thresholds $b_0, \dots, b_k$ for any k -valued threshold function $f(x_1, \dots, x_n)$ given by its values $f(c_1, \dots, c_n)$ for all $(c_1 \dots c_n)$ in $\{0, \dots, k-1\}^n$.

In computer experiment they showed the convergence of this algorithm on many different examples. Here, we present a theoretical proof of this algorithm convergence on each k -valued threshold function for a finite number of steps (iterations). The proof is very much similar to the geometrical proof of perceptron convergence theorem by M. Minsky and S. Papert.

Keywords: *threshold functions, iterative algorithms, convergence.*

Введение

В целом ряде разделов дискретной математики возникает задача распознавания принадлежности функции к классу пороговых и восстановления (характеризации) неизвестной дискретной функции из этого класса с помощью последовательных вопросов о её значениях в точках.

Определение 1 [1]. Функция k -значной логики $f(x_1, \dots, x_n)$, для которой существует линейная форма $L(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$, $x_i \in \mathbb{Z}_k$ с вещественными коэффициентами и набор вещественных порогов $b_0 < b_1 < \dots < b_k$, такие, что для всех $i \in \{0, \dots, k-1\}$ выполняется условие

$$f(x_1, \dots, x_n) = i \iff b_i \leq L(x_1, \dots, x_n) < b_{i+1},$$

называется *пороговой k -значной функцией*.

Замечание 1. В силу неоднозначности задания пороговой функции будем полагать возможным использование порогов, удовлетворяющих нестрогому неравенству

$$b_0 \leq b_1 \leq \dots \leq b_k.$$

В случае равенства порогов $b_i = b_{i+1}$ для некоторого $i \in \{0, \dots, k-1\}$, очевидно, функция $f(x_1, \dots, x_n)$ не принимает значения i . Если пороговая функция принимает значение i , то будем считать, что выполняется строгое двухстороннее неравенство

$$f(x_1, \dots, x_n) = i \iff b_i < L(x_1, \dots, x_n) < b_{i+1}.$$

Этого всегда можно добиться небольшим изменением порогов или весов.

Под алгоритмом характеристики пороговой k -значной функции понимается процедура нахождения какого-либо семейства параллельных гиперплоскостей, разделяющих множества различных значений данной функции, то есть нахождения коэффициентов $a_1, a_2, \dots, a_n$ линейной формы $L(x_1, \dots, x_n)$ и множества порогов $b_0, b_1, \dots, b_{k-1}$.

Подобная задача рассматривалась в работе [2], где предложен итеративный алгоритм нахождения представления k -значной пороговой функции, который на практических примерах показал сходимость.

Далее описан новый алгоритм характеристики k -значных пороговых функций и доказана его сходимость. Как показывают эксперименты [1], трудоёмкость нового алгоритма на порядки меньше, чем у алгоритма в [2].

1. Основные понятия и характеристики пороговых функций

Определение 2. Будем говорить, что линейная форма $L(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$ даёт *чистое разделение* областей значений функции $f(x_1, \dots, x_n)$, принимающей все значения из $\mathbb{Z}_k$, если для любого $\alpha \in \{0, \dots, k-2\}$ выполняется строгое неравенство

$$\max_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha} \{a_1\varepsilon_1 + \dots + a_n\varepsilon_n\} < \min_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha+1} \{a_1\varepsilon_1 + \dots + a_n\varepsilon_n\}.$$

Если это выполняется, то границы $b_0, b_1, \dots, b_k$ можно определить, например, следующим способом:

$$b_\alpha = \min_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\}, \quad \alpha \in \mathbb{Z}_k,$$

$$b_k = \max_{f(\varepsilon_1, \dots, \varepsilon_n) = k-1} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\} + 1.$$

Замечание 2. В случае, когда функция $f(x_1, \dots, x_n)$ не принимает некоторых значений из $\mathbb{Z}_k$, необходимо следующим образом убрать из рассмотрения соответствующие области значений: пусть функция $f(x_1, \dots, x_n)$ принимает значения $0 \leq \alpha_0 < \alpha_1 < \dots < \alpha_t < k$, $0 < t < k$. Тогда для всех $i \in \{0, \dots, t-1\}$ необходимо проверить выполнение строгого неравенства

$$\max_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha_i} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\} < \min_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha_{i+1}} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\}.$$

Пороги в этом случае можно определить, например, следующим образом: для всех значений $\alpha_0, \dots, \alpha_t$ присвоить значения

$$b_{\alpha_i} = \min_{f(\varepsilon_1, \dots, \varepsilon_n) = \alpha_i} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\}; \quad (1)$$

далее, если функция принимает значение $k-1$, то положить

$$b_k = \max_{f(\varepsilon_1, \dots, \varepsilon_n) = k-1} \{a_1 \varepsilon_1 + \dots + a_n \varepsilon_n\} + 1, \quad (2)$$

в противном случае положить $b_k = +\infty$; для оставшихся значений $j \in \mathbb{Z}_k \setminus \{\alpha_0, \dots, \alpha_t\}$ (которые не являются значениями функции $f(x_1, \dots, x_n)$), начиная со старшего, присвоить соответствующему порогу значение $b_j = b_{j+1}$.

В работе [1] представлен подробный обзор существующих методов характеристики k -значных пороговых функций и предложен новый алгоритм характеристики, использующий в качестве первичной аппроксимации коэффициентов линейной формы коэффициенты роста и коэффициенты возрастания.

Определение 3. Для функции k -значной логики $f(x_1, \dots, x_n)$ коэффициентом роста по переменной x_i называется величина

$$\Delta_i = \sum_{(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)} (f(x_1, \dots, x_{i-1}, k-1, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)).$$

Определение 4. Для функции k -значной логики $f(x_1, \dots, x_n)$ коэффициентом возрастания по переменной x_i называется величина

$$\lambda_i = \sum_{(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \mathbb{Z}_k^{n-1}} \sum_{l=0}^{k-2} \sum_{\varepsilon=l+1}^{k-1} (f(x_1, \dots, x_{i-1}, \varepsilon, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, l, x_{i+1}, \dots, x_n)).$$

2. Алгоритм характеристики k -значных пороговых функций

Приведём новый алгоритм (алгоритм 1) характеристики k -значных пороговых функций в формализованном виде. Для этого понадобятся следующие обозначения для произвольного $i \in \mathbb{Z}_k$:

$$F_i = \{(x_1, \dots, x_n) \in \mathbb{Z}_k^n : f(x_1, \dots, x_n) = i\};$$

$$\max(F_i) = \max_{(x_1, \dots, x_n) \in F_i} \{L(x_1, \dots, x_n)\}; \quad (3)$$

$$x_{\max}(F_i) = \{(x_1, \dots, x_n) \in \mathbb{Z}_k^n : L(x_1, \dots, x_n) = \max(F_i)\};$$

$$\min(F_i) = \min_{(x_1, \dots, x_n) \in F_i} \{L(x_1, \dots, x_n)\}; \quad (4)$$

$$x_{\min}(F_i) = \{(x_1, \dots, x_n) \in \mathbb{Z}_k^n : L(x_1, \dots, x_n) = \min(F_i)\}.$$

В случае если $F_i = \emptyset$, положим $\max(F_i) = +\infty$ и $\min(F_i) = -\infty$, $x_{\min}(F_i) = x_{\max}(F_i) = (-1, \dots, -1)$.

Пусть пороговая функция $f(x_1, \dots, x_n)$ принимает только значения $0 \leq \alpha_0 < \alpha_1 < \dots < \alpha_t < k$, $0 < t < k$.

Алгоритм 1. Новый алгоритм характеристики пороговых k -значных функций

Вход: $\Delta_i, \lambda_i, i = 1, \dots, n; F_{\alpha_i}, \alpha_i, i = 0, \dots, t$

// Инициализация коэффициентов линейной формы

1: Положить $(a_1, \dots, a_n) := (\Delta_1, \dots, \Delta_n)$ либо $(a_1, \dots, a_n) := (\lambda_1, \dots, \lambda_n)$

// Проверка чистого разделения областей значений функции

2: Для всех $i = 0, \dots, t-1$

3: вычислить $\min(F_{\alpha_{i+1}})$ по (4);

4: вычислить $\max(F_{\alpha_i})$ по (3).

5: Если $\max(F_{\alpha_i}) \geq \min(F_{\alpha_{i+1}})$, то

// Блок «Коррекция»

6: $x \stackrel{\mathcal{R}}{\leftarrow} x_{\max}(F_{\alpha_i})$,

7: $y \stackrel{\mathcal{R}}{\leftarrow} x_{\min}(F_{\alpha_{i+1}})$.

8: Для всех $j = 1, \dots, n$

9: $a_j := a_j - x_j + y_j$. // Коррекция коэффициентов линейной формы

10: Перейти на шаг 2.

// Блок «Вычисление порогов»

11: Вычислить $\min(F_{\alpha_0})$.

12: Для всех $i = 0, \dots, t$

13: положить $b_{\alpha_i} := \min(F_{\alpha_i})$ // в соответствии с (1).

14: Если $\alpha_t = k-1$, то

15: вычислить $\max(F_{k-1})$,

16: положить $b_k := \max(F_{k-1}) + 1$, // в соответствии с (2)

17: иначе

18: $b_k := +\infty$.

// Задание остальных порогов (см. замечание 2)

19: $Z := \mathbb{Z}_k \setminus \{\alpha_0, \dots, \alpha_t\}$.

20: Пока $Z \neq \emptyset$

21: $j := \max Z$, $b_j = b_{j+1}$, $Z = Z \setminus \{j\}$.

22: Вывести $(a_1, \dots, a_n), (b_0, \dots, b_k)$

На шаге 1 вектор коэффициентов линейной формы инициализируется коэффициентами роста либо коэффициентами возрастания. На шагах 6 и 7 производится случайный и равновероятный выбор точек x и y из множеств $x_{\max}(F_{\alpha_i})$ и $x_{\min}(F_{\alpha_{i+1}})$ соответственно.

3. Доказательство сходимости

На практике алгоритм показал сходимость во всех рассмотренных случаях при изменении параметров в широких интервалах. Более того, в сравнении с известным алгоритмом Обрадовича [2] он демонстрирует значительное превосходство по трудоёмкости, а на параметрах n и k больше 10 — на несколько порядков. В связи с этим теоретическое доказательство его сходимости можно рассматривать как ожидаемое и чрезвычайно важное для данной области исследования. Предварительно отметим, что доказательство теоремы 1 во многом схоже с геометрическим доказательством теоремы о сходимости персептрона, изложенным в работе [3].

Теорема 1. Если f — k -значная пороговая функция, то алгоритм 1 сходится за конечное число шагов и даёт её характеристику.

Доказательство. Пусть пороговая функция k -значной логики $f(x_1, \dots, x_n)$ задаётся линейной формой $L(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$, т.е. вектором $\mathbf{L} = (a_1, a_2, \dots, a_n)$. Вокруг вектора $\mathbf{L}$ всегда можно построить конус допустимых решений Q , такой, что любой вектор, лежащий в данном конусе, даёт чистое разделение областей значений функции $f(x_1, \dots, x_n)$ [3].

Конусом допустимых решений Q назовём множество векторов, для которых выполнены следующие свойства:

- 1) $\alpha \in Q \implies k\alpha \in Q$ для всех $k > 0$;
- 2) $\alpha \in Q, \beta \in Q \implies (\alpha + \beta) \in Q$;
- 3) для любого $\alpha \in Q$ линейная форма, построенная по коэффициентам вектора α , даёт чистое разделение областей значений функции $f(x_1, \dots, x_n)$.

Обозначим угол раствора конуса Θ ($\Theta > 0$). Докажем, что за конечное число шагов алгоритма вектор коэффициентов линейной формы окажется внутри конуса допустимых решений.

Очевидно, что в силу абсолютной монотонности пороговой функции [4] векторы коэффициентов роста и коэффициентов возрастания лежат в одном ортанте с вектором $\mathbf{L}$. Таким образом, алгоритм начинает работу с вектора $\mathbf{A}_0$, лежащего в одном ортанте с вектором $\mathbf{L}$.

Рассмотрим операцию коррекции в алгоритме. Обозначим текущий вектор линейной формы $\mathbf{A}_i$, он поступает на вход блока «Коррекция», после коррекции переходит в вектор $\mathbf{A}_{i+1}$.

Алгоритм осуществляет коррекцию с помощью двух точек $\mathbf{u}_i \in x_{\max}(F_{\alpha_j})$ и $\mathbf{v}_i \in x_{\min}(F_{\alpha_{j+1}})$ тогда и только тогда, когда одновременно выполняются следующие два неравенства:

- 1) $\mathbf{A}_i\mathbf{v}_i \leq \mathbf{A}_i\mathbf{u}_i$, так как задействован блок «Коррекция» алгоритма;
- 2) при этом для неизвестного истинного вектора $\mathbf{L}$ выполнено противоположное неравенство $\mathbf{L}\mathbf{v}_i > \mathbf{L}\mathbf{u}_i$ в силу определения 1.

Эти неравенства равносильны системе

$$\begin{cases} \mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > 0, \\ \mathbf{A}_i(\mathbf{v}_i - \mathbf{u}_i) \leq 0. \end{cases} \quad (5)$$

Вектор коррекции $(\mathbf{v}_i - \mathbf{u}_i) \equiv \mathbf{C}_i$ прибавляется к текущему значению $\mathbf{A}_i$ и получается новое значение $\mathbf{A}_{i+1} = \mathbf{A}_i + \mathbf{C}_i$.

Рассмотрим плоскость, на которой лежат векторы $\mathbf{L}$ и $\mathbf{A}_i$ (рис. 1). Обозначим $\mathbf{C}_{LA_i}$ проекцию вектора $\mathbf{C}_i$ на плоскость, образованную векторами $\mathbf{L}$ и $\mathbf{A}_i$. Перенесём начало этой проекции в конец вектора $\mathbf{A}_i$.

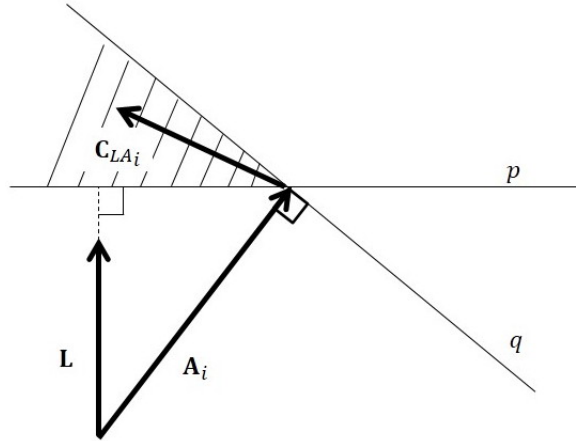


Рис. 1. Проекция вектора коррекции

В силу первого неравенства системы (5) конец вектора $\mathbf{C}_{LA_i}$ должен быть ниже прямой q , в силу второго неравенства — выше прямой p , т. е. должен лежать в заштрихованной области на рис. 1. Таким образом, вектор $\mathbf{C}_{LA_i}$ направлен к вектору $\mathbf{L}$.

Рассмотрим конус K_i , образованный вращением вектора $\mathbf{A}_i$ вокруг вектора $\mathbf{L}$. Видно, что вектор $\mathbf{C}_i$ заходит внутрь конуса K_i . Для каждого шага алгоритма возможны два случая: вектор $\mathbf{C}_i$ лежит внутри конуса K_i целиком или выходит за пределы конуса K_i , то есть «прокалывает» конус, как на рис. 2.

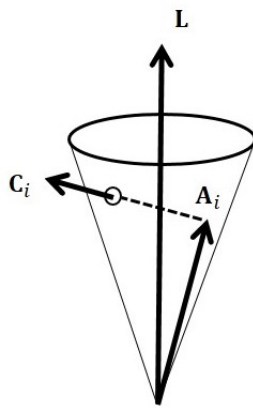


Рис. 2. «Прокол конуса»

Покажем, что «прокол конуса» может произойти конечное число раз. Так как по замечанию 1 для любых $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{Z}_k^n$ выполняется неравенство $\mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > 0$, существует $r > 0$, такое, что $\mathbf{L}(\mathbf{v}_i - \mathbf{u}_i) > r$ для любых $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{Z}_k^n$. Это означает, что часть заштрихованной на рис. 1 области, заведомо содержащая конец вектора $\mathbf{C}_{LA_i}$, лежит на $r/|\mathbf{L}|$

выше, чем окончание вектора $\mathbf{A}_i$. Рис. 3 является модификацией рис. 1, учитывающей вышеизложенное. Если рассмотреть проекцию на плоскость основания конуса K_i , то расстояние от конца проекции на эту плоскость вектора $\mathbf{C}_i$ до касательной к окружности конуса K_i в точке окончания вектора $\mathbf{A}_i$ будет не менее чем d .

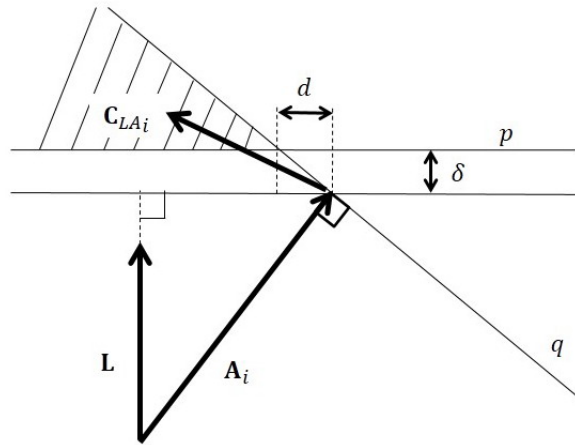


Рис. 3. Проекция вектора коррекции

Нижнюю границу для d определяет тот факт, что угол между векторами $\mathbf{A}_i$ и $\mathbf{L}$ составляет не меньше Θ (иначе алгоритм останавливает работу) и меньше $(\pi/2 - \Theta)$; таким образом, $d \geq (r/|L|) \operatorname{ctg} \Theta > 0$. Длина вектора $\mathbf{C}_i$ ограничена сверху величиной $(k-1)\sqrt{n}$.

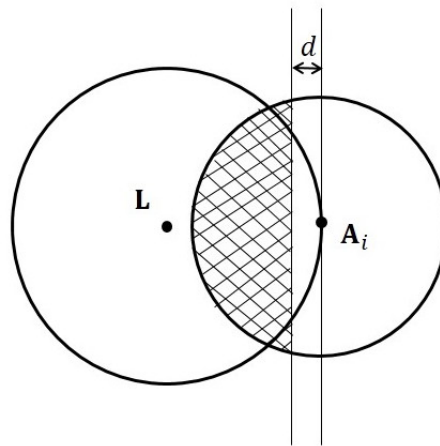
Рассмотрим проекцию на плоскость основания конуса K_i . На рис. 4 изображены две окружности: окружность основания конуса K_i вокруг вектора $\mathbf{L}$ и окружность радиуса $(k-1)\sqrt{n}$ вокруг окончания вектора $\mathbf{A}_i$. К окружности основания конуса K_i проведена касательная в точке окончания вектора $\mathbf{A}_i$. На расстоянии d от касательной проведена линия.

Согласно вышеизложенному, окончание проекции вектора $\mathbf{C}_i$ на данную плоскость должно лежать на расстоянии не меньше d от касательной, не больше $(k-1)\sqrt{n}$ от окончания вектора $\mathbf{A}_i$ и по одну сторону с вектором $\mathbf{L}$ относительно касательной. Таким образом, окончание проекции вектора $\mathbf{C}_i$ лежит в заштрихованной на рис. 4 области.

Можно легко заметить, что заштрихованная область не целиком лежит внутри окружности основания конуса K_i . Когда окончание проекции вектора $\mathbf{C}_i$ будет попадать в заштрихованную область вне окружности основания конуса K_i , будет происходить «прокол конуса».

Покажем, что с ростом числа итераций i «прокол конуса» прекратится. Так как вектор $\mathbf{C}_i$ всегда содержит вертикальную составляющую не менее $r/|L|$, высота конуса K_i растет с каждой итерацией алгоритма. Угол между векторами $\mathbf{L}$ и $\mathbf{A}_i$ больше Θ (в противном случае алгоритм останавливается и решение найдено). Таким образом, радиус основания конуса K_i увеличивается неограниченно.

С увеличением радиуса основания конуса K_i вся заштрихованная область попадет внутрь окружности основания конуса K_i , как показано на рис. 5, и «прокол конуса» прекратится. Таким образом, «прокол конуса» может произойти конечное число раз, после чего вектор коррекции $\mathbf{C}_i$ целиком будет попадать в конус K_i и оконча-

Рис. 4. Проекция на плоскость основания конуса K_i

ние вектора коррекции будет всегда попадать внутрь круга фиксированного радиуса. С ростом числа итераций высота конуса K_i растёт неограниченно минимум на $r/|L|$ с каждой итерацией (см. рис. 3). Вместе с высотой конуса растёт радиус основания конуса допустимых решений Q — минимум на величину $r \cos \Theta/|L|$.

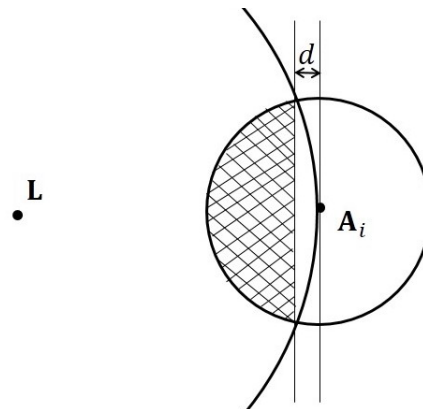


Рис. 5. Проекция на плоскость основания конуса

Итак, начиная с некоторого шага алгоритма, окончания всех векторов A_i будут лежать в вертикальном цилиндре, расположенном внутри конуса допустимых решений, имеющего своей осью вектор L . ■

Заключение

Предложенный в работе [1] алгоритм сходится и находит за конечное число шагов решение проблемы характеристики k -значных пороговых функций.

Сходимость алгоритма 1, как следует из доказательства, выполняется при использовании и коэффициентов роста, и коэффициентов возрастания для первичной аппроксимации коэффициентов линейной формы. Вместе с тем эксперименты показывают, что получаемые в этих двух случаях результаты, а также сложности реализации алгоритмов отличаются и требуют дальнейшего изучения.

ЛИТЕРАТУРА

1. Бурделёв А. В., Никонов В. Г. О новом алгоритме характеристики k -значных пороговых функций // Computational Nanotechnology. 2017. Вып. 1. С. 7–14.
2. Obradovic Z. and Parberry I. Learning with discrete multi-valued neurons // Proc. 7th Intern. Conf. Machine Learning. University of Texas, Austin, Texas, June 21–23 1990. P. 392–399.
3. Минский М., Пейперт С. Перцептроны. М.: Мир, 1971.
4. Никонов В. Г., Никонов Н. В. Особенности пороговых представлений k -значных функций // Труды по дискретной математике. 2008. Т. 11. Вып. 1. С. 60–85.

REFERENCES

1. Burdelev A. V. and Nikonov V. G. O novom algoritme harakterizacii k -znachnyh porogovyh funkcii [A new algorithm for recognition of k -valued threshold functions]. Computational Nanotechnology, 2017, vol. 1, pp. 7–14. (in Russian)
2. Obradovic Z. and Parberry I. Learning with discrete multi-valued neurons. Proc. 7th Intern. Conf. Machine Learning, University of Texas, Austin, Texas, June 21–23 1990, pp. 392–399.
3. Minsky M. and Papert S. Perceptrons. Cambridge, MA, MIT Press, 1969.
4. Nikonov V. G. and Nikonov N. V. Osobennosti porogovyh predstavlenii k -znachnyh funkcii [Features of threshold representations of k -valued functions]. Tr. Diskr. Mat., 2008, vol. 11, iss. 1, pp. 60–85. (in Russian)