МАШИННЫЕ МЕТОДЫ ОБНАРУЖЕНИЯ ЗАКОНОМЕРНОСТЕЙ
(Вычислительные системы)
1981 год Выпуск 88

УДК 578.087.1:519.9

# МЕТОДЫ ПОСТРОЕНИЯ АЛГОРИТМОВ ЭТАЛОННОГО ТИПА В ЗАДАЧАХ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ

В.М.Бухштабер, В.К.Маслов, Е.А.Зеленюк

Автоматическая классификация является одним из важнейших разделов анализа данных [1,6,12]. Большинство методов и алгоритмов автоматической классификации возникли из конкретных задач в различных областях прикладных исследований и носят отпечаток языка и традиций этих областей. Сейчас общепризнано, что для дальнейшего развития тематики автоматической классификации необходима классификация самих алгоритмов автоматической классификации [2,12] прежде всего для сравнительного анализа известных и конструирования новых алгоритмов.

Предлагаемый в [4] подход к построению теории автоматической классификации основан на исследовании математических моделей алгоритмов и позволяет разработать средство их единообразного описания, систематизации и сравнения. Выделяемое при этом понятие движения алгоритма становится самостоятельным объектом исследования. На основе описания движения алгоритма удается глубже оценить результаты работы алгоритмов и строить алгоритмы с заранее заданными свойствами. Настоящая статья посвящена изложению и развитию этого подхода в рамках важного класса - алгоритмов эталонного типа [1].

## §1. Структура алгоритмов автоматической классификации

В структуре алгоритмов мы выделяем следующие элементы (их содержательный смысл будет разъяснен на примерах): $X$ - выборка объектов, представленная в виде множества точек пространства наблюдений; $P(X)$ - множество выделенных конечных подмножеств в $X$; $S(X)$ - множество состояний, в которых выборка $X$ участвует в ал-

горитме; $E(X)$ - множество описаний выборки $X$, допускаемых данным алгоритмом; $G$ - оператор $P \times S \times E \times N \to P$, где $N$ - множество натуральных чисел; $K$ - оператор $S \times E \times P \to S$; $D$ - оператор $E \times S \to E$. Этих элементов достаточно для описания итераций алгоритма, или, как мы будем говорить, движения алгоритма. А именно, выбрав начальные данные $s_1 \in S$, $e_1 \in E$ и $p_1 \in P$, мы получаем последовательность $(s_1,e_1),\ldots,(s_n,e_n),\ldots,(s_{n+1},e_{n+1}),\ldots$, где $s_{n+1} = K(s_n,e_n,p_n)$, $e_{n+1} = D(e_n,s_{n+1})$, $p_{n+1} = G(p_n,s_n,e_n,n)$.

Оператор $G$ естественно назвать генератором, так как его значением является порция объектов $p_n$, поступающая на $n$-м шаге алгоритма.

Для того чтобы пояснить роль оператора $K$, необходимо уточнить взаимоотношения между выборкой $X$ и множеством ее состояний $S = S(X)$. Мы будем предполагать, что каждому вложению $I'\colon X' \subset X$ соответствует отображение $S[I']\colon S(X) \to S(X')$, причем:

1) если $I'\colon X' \subset X$ разлагается в композицию $X' \subset X'' \subset X$, то и отображение $S[I]\colon S(X) \to S(X')$ разлагается в композицию $S(X) \to S(X'') \to S(X')$;

2) если $I'\colon X' \subset X$, $I''\colon X'' \subset X$ - вложение двух подмножеств $X'$, $X''$ и $X' \cup X'' = X$, то для любых $s_1, s_2 \in S$ из равенств $S[I'](s_1) = S[I'](s_2)$, $S[I''](s_1) = S[I''](s_2)$ следует равенство $s_1 = s_2$.

Для данного состояния $s = s(X)$ и вложения $I\colon X' \subset X$ состояние $S[I](s(X))$ выборки $X$ называется индуцированным состоянием. Таким образом, для каждого состояния $s(X)$ определено отображение, сопоставляющее объекту $x$ индуцированное состояние. Это отображение будем называть $s$-классификацией. Учитывая изложенное, оператор $K$ естественно назвать классификатором. Анализ аксиом 1 и 2 (см. выше) показывает, что множество состояний $S$ обязано лежать в множестве отображений из $X$ в $Z$, $S \subset \{X \to Z\}$, где $Z$ - множество возможных значений $s$-классификаций. Таким образом, для задания множества $S(X)$ достаточно построить множество $Z$ и указать ограничения, выделяющие $S$ в множестве $\{X \to Z\}$. С другой стороны, из самой роли множества $Z$ вытекает, что для его построения можно использовать всю информацию о структуре выборки $X$ в пространстве наблюдений.

Без дополнительных объяснений понятно, что оператор $D$ можно назвать дескриптором.

ОПРЕДЕЛЕНИЕ 1. Алгоритм автоматической классификации называется сходящимся, если в процессе его движения $(s_1,e_1),\ldots,(s_n,e_n),\ldots$ последовательность описаний $e_1,\ldots,e_n,\ldots$ сходится в мет-

рике пространства E к некоторому предельному описанию $e$. Будем говорить, что движение алгоритма стабилизируется, если, начиная с некоторого номера $n_1$, выполняются равенства $e_n = e_{n+1}, s_n(x) = s_{n+1}(x)$ для всех $n \geq n_1$, $x \in p_n \subset X$.

ОПРЕДЕЛЕНИЕ 2. Моделью алгоритма автоматической классификации называется набор $(X; P(X), S(X), E(X); G, K, D; p_1, s_1, e_1)$ элементов структуры алгоритма, удовлетворяющих указанным выше предположениям, и начальных данных $p_1 \in P$, $s_1 \in S$, $e_1 \in E$.

ОПРЕДЕЛЕНИЕ 3. Функционал $F: S \times E \to R^1$ называется интерпретирующим для модели алгоритма, если

$$F(s_n, e_n) \geq F(s_{n+1}, e_n), \quad F(s_{n+1}, e_n) \geq F(s_{n+1}, e_{n+1}) \qquad (1)$$

для всех $n$, начиная с некоторого $n_0$.

Интерпретирующий функционал является средством для исследования движения алгоритма. Выбор из множества функционалов так называемого целевого функционала, дающего содержательную интерпретацию предельного описания, требует привлечения дополнительных содержательных соображений. Это означает, что задачи исследования движения и интерпретации цели движения не совпадают, но дополняют друг друга.

Проиллюстрируем содержательный смысл описанного способа построения основных элементов структуры алгоритма автоматической классификации на примерах.

## §2. Модели алгоритмов типа "Форель"

Рассмотрим описание широко известного алгоритма "Форель" [6], исследованию которого были посвящены работы [9-11].

ПРИМЕР 1. Алгоритм "Форель". Пусть $X$ - множество точек евклидова пространства $R^J$; $P(X)$ состоит из одного элемента $[X]$, обозначающего всю выборку $X$; $S(X)$ - множество всех подмножеств $X$; множество описаний $E(X) \equiv R^J$; $G$ - постоянное отображение, $G(p, s, e, n) \equiv [X]$; $K(s, e, [X]) = s' = \{x \in X: |x - e| \leq r\}$, где $r$ - фиксированное число, $|x-e| = \left(\sum_{j=1}^{J} |x(j)-e(j)|^2\right)^{1/2}$. $D(e,s) = e' = \arg \min_{e \in R^J} \min_{x \in s} |x-e|^2$; очевидно, что $e' = \frac{1}{|s|} \sum_{x \in s} x$, где $|s|$ - количество точек в множестве $s$, т.е. $e'$ - центр тяжести множества $s$.

Тогда $F(s,e) = \sum_{x \in s} (|x-e|^2 - r^2)$ - интерпретирующий функционал. Неравенства (1) для него были получены в [9] и использовались для доказательства сходимости алгоритма "Форель". Некоторые обобщения алгоритма "Форель", связанные с введением различных метрик $\rho(x,e)$ и интерпретирующих функционалов, предлагались в работе [10].

Применение алгоритма "Форель" опирается на следующую гипотезу - выборка X представляет собой объединение $X^1 \cup X^2$, где $X^1$ - компактный сгусток, в котором ядро совпадает с геометрическим центром, а точки $X^2$ лежат на достаточном удалении от сгустка. Искомым предельным состоянием $s_*$ является классификация выборки X, выделяющая в X подмножество $X^1$, или, другими словами, определяющая для каждой точки $x \in X$ принадлежность ее к множеству $X^1$; при четкой классификации для определения такой принадлежности достаточно 0 и 1, т.е. $s_* = X \to \{0,1\}$. Будем считать, что значения $s_*(x)$ для всех $x \in X$ задаются некоторой функцией принадлежности $\lambda(e,x)$; при этом гипотеза "Форели" о виде сгустка отразится на построении этой функции следующим образом: совпадение ядра сгустка с геометрическим центром определяет равенство значений $\lambda(e,x)$ внутри сгустка, а "достаточная удаленность" точек $X^2$ позволяет использовать в качестве меры удаленности число r для отделения $X^1$ от $X^2$, т.е.

$$\lambda(e,x) = \gamma'(|e-x|) \qquad \begin{cases} 1, |e-x| \le r, \\ 0, |e-x| > r. \end{cases}$$

Опираясь на приведенный анализ гипотезы, можно дать описание алгоритма "Форель" в терминах функции принадлежности $\lambda(e,x)$, взяв в качестве множества возможных значений s-классификаций множество $Z = \{0,1\}$. Оно отличается от данного выше описания определяющими множества состояний $S \equiv \{X \to \{0,1\}\}$, дескриптора

$$D(e,s) = \sum_{x \in X} \frac{\gamma(|e-x|)}{\sum_{x \in X} \gamma(|e-x|)} x$$

и классификатора $K(s,e,[X])(x) = \gamma'(|e-x|)$.

Различные гипотезы о виде сгустка, выраженные в терминах функций принадлежности $\lambda(e,x)$, задают семейство алгоритмов типа "Форель". Дадим описание алгоритмов этого семейства. В качестве Z возьмем множество значений функции принадлежности $\lambda$: $Ex\, X \to [0,1]$, т.е. $Z \subset [0,1]$.

X, E(X), P(X), и G - те же, что и в примере I. Множество состояний $S(X) = \{X \to Z\}$. Операторы K и D определяются так:

$$K(s,e,[X])(x) = K(e)(x) = s'(x) = \lambda(e,x) = \gamma(|e-x|),$$

$$D(e,s) = D(s) = e' = \sum_{x \in X} \frac{\gamma(|e-x|)}{\sum_{x \in X} (|e-x|)}.$$

Как видим, дескриптор D задает взвешенный центр тяжести точек $x \in X$, где веса пропорциональны значениям функции принадлежности, определяющей классификацию $s'$, т.е. D определяет положение текущего ядра сгустка, исходя из предположения о структуре выборки в окрестности ядра искомого сгустка, выражаемых функцией $\lambda(e, x)$. Это могут быть предположения, например, о наличии в выборке выбросов, смещающих ядро сгустка, - тогда действие функции $\lambda(e,x)$ направлено на устранение такого смещения, что аналогично действию $\lambda$-весов [8], или о том, что множество наблюдений в окрестности ядра есть "размытое множество" [I2] - тогда $\lambda(e,x)$, определяя принадлежность точки к сгустку, "фокусирует" центр $e'$ в ядре сгустка; возможны и другие гипотезы.

## §3. Эталонные и базисные эталонные алгоритмы

Термин "эталонные алгоритмы" взят из [I]. Под алгоритмом эталонного типа (эталонным алгоритмом) будем понимать алгоритм автоматической классификации, описываемый моделью в смысле определения I со следующей конкретизацией структуры: X и P(X) - те же, что в общем случае, S(X) - множество состояний (множество s-классификаций), определяющих разбиение выборки X на k непересекающихся подмножеств, $k \in [k_{min}, k_{max}]$. Подчеркнем, что в общем случае множество S не обязано совпадать с множеством разбиений выборки X на k подмножеств (так, элементы $s \in S$ могут содержать еще информацию о структуре отдельных классов). E(X) - множество описаний выборки X, точками которого являются наборы $\bar{e} = (e^1, \ldots, e^l, \ldots) \in E$, причем эталон l-го класса $e^l$ принадлежит своему пространству эталонов $E^l$, т.е. $e^l \in E^l$, $E = E^1 x \ldots x E^l x \ldots$. Конкретизация оператора K для эталонных алгоритмов состоит во введении набора функционалов $\mu_l : X \times E^l \to R^1$, характеризующих для каждой пары $(x, e^l)$ "меру несходства" при замене объекта x эталоном $e^l$,

$$K(s,\bar{e},p)(x) = \begin{cases} X^1, \text{если } x \in p \text{ и } \mu_1(x,e^1) \le \min\limits_{l \ge 1} \mu_l(x,e^l), \\ X^{l'}, \text{если } x \in p \cap (X \setminus \bigcup\limits_{l=1}^{l'-1} X^l) \text{ и} \\ \quad \mu_{l'}(x,e^{l'}) \le \min\limits_{l \ge 1} \mu_l(x,e^l). \end{cases} \qquad (2)$$

Конкретизация оператора D в эталонном алгоритме не проводится; она, как и в общем случае, определяется структурой множеств состояний и описаний.

Рассмотрим отображение $M_l: E^l \times S \to R^1$, $M_l(e^l, s) = \sum_{x \in X^l} \varphi(\mu_l(x, e^l))$, где $X^l$ - l-й класс, определяемый состоянием s, а $\varphi$ - строго монотонно возрастающая функция, $\varphi(0) = 0$. Положим

$$D(e,s) = (e^1_*, \ldots, e^l_*, \ldots), \quad e^l_* = \arg\min_{e \in E^l} M_l(e,s). \quad (3)$$

ОПРЕДЕЛЕНИЕ 4. Базисным эталонным алгоритмом будем называть алгоритм эталонного типа, в модели которого операторы K и D задаются формулами (2) и (3) соответственно.

Непосредственно из формул (2) и (3) следует, что функционал

$$F(s,e) = \sum M_l(e^l, s) \quad (4)$$

является интерпретирующим для модели базисного эталонного алгоритма.

Типичным примером базисного эталонного алгоритма является широко известный алгоритм k-средних [I2].

ПРИМЕР 2. Алгоритм k-средних. X, P(X) и G - те же, что в примере I. S(X) - множество разбиений X на k непересекающихся подмножеств, т.е. $s = (X^1, \ldots, X^k)$, $\bigcup_{l=1}^{k} X^l = X$, $X^{l'} \cap X^{l''} = \emptyset$ для $l' \neq l''$. E(X) - множество наборов из k точек, $\bar{e} = (e^1, \ldots, e^k)$, $e^l \in R^J$. Метрика в E вводится как в пространстве $R^J \times \ldots \times R^J$, а именно $\rho(\bar{e}_1, \bar{e}_2) = \left(\sum_{l=1}^{k} |e^l_1 - e^l_2|^2\right)^{1/2}$. $K(s, \bar{e}, [X]) = s_* = \{X^l_*\}_{l=1}^k$, где $X^1_* = \{x \in X: |x - e^1| = \min_{1 \le l' \le k} |x - e^{l'}|\}$, $X^l_* = \{x \in X \setminus \bigcup_{l'=1}^{l-1} X^{l'}_* : |x - e^l| = \min_{1 \le l' \le k} |x - e^{l'}|\}$, $l > 1$; $D(\bar{e}, s) = (e^1_*, \ldots, e^k_*)$, где $e^l_* = \arg\min_{e^l} \sum_{x \in X^l} |x - e^l|^2$.

Положим $F(s,e) = \sum_{l=1}^{k} \left(\sum_{x \in X^l} |x - e^l|^2\right)$. Непосредственно из описания K и D следует, что F - интерпретирующий функционал, а функции $\mu_l$ и $M_l$ можно определить так: $\mu_l = |x - e^l|$, $M_l = \sum_{x \in X^l} |x - e^l|^2$.

Свобода в выборе пространств $E_1,\dots,E_l,\dots$ и функционалов $\mu_1,\dots,\mu_l,\dots$ позволяет в рамках модели базисного эталонного алгоритма охватить широкий круг моделей алгоритмов автоматической классификации, в том числе алгоритмы: "Форель", k-средних, "эталонные алгоритмы" в смысле [I] и алгоритмы, получаемые по "методу динамических сгустков" ( MNDS, [5]).

Для любого алгоритма эталонного типа можно ввести модификацию, заключающуюся во введении дополнительного класса "джокер" ("отказ", не знаю и т.п.). Обычно для этого задается "порог отказа" $\delta$, и если значения мер несходства $\mu_l$ превосходят порог $\delta$ для всех $l$, то оператор K относит такой объект к символическому эталону • класса "джокер". Примеры использования класса "джокер" будут даны ниже.

Рассмотрим множество Z, для построения которого, как сказано в §I, можно использовать всю априорную информацию о структуре выборки X в пространстве наблюдений. Систематизация имеющейся информации позволяет представить множество Z в виде объединения непересекающихся подмножеств $Z^l$, $Z = \bigcup_{l=1}^{k} Z^l$, где $Z^l$ - множество $z^l_1,\dots,z^l_{m_l}$, учитывающее априорную информацию о l-м классе. Поэтому каждое состояние $s \in S$ определяет разбиение выборки X на k непересекающихся подмножеств $X^l = s^{-1}(Z^l)$. Поскольку описание выборки должно быть согласовано с априорной информацией о ней, структура множества описания должна быть функций множества Z. Эти рассуждения позволяют еще более конкретизировать структуру множества описаний E в универсальном алгоритме эталонного типа. Будем считать, что $E = E^1 \times \dots \times E^l \times \dots$, где $E^l = E(z^l_1) \times \dots \times E(z^l_{m_l})$ и $E(z^l_i)$ - пространство эталонов, соответствующее элементу $z^l_i$, характеризующему i-ю априорную информацию об l-м классе. Такое представление пространств $E^l$ дает возможность использовать элементы $z^l_i$ для целенаправленного конструирования функционалов $\mu_l$ с использованием априорной информации о структуре выборки в пространстве наблюдений. Проиллюстрируем сказанное на примере.

ПРИМЕР 3. Есть гипотеза, что выборку данных X в пространстве наблюдений можно представить в виде объединения $X = X^1 \cup X^2$, где $X^1$ - сгусток сложной формы, для описания которого достаточно k эталонов $(e_1,\dots,e_k)$; при этом любой объект $x \in X^1$ имеет меру несходства $\mu(x) = \min_l \mu_l(x, e^l) \le r$, где r - фиксированное число, иг-

рающее роль порога отказа. Тогда $Z = \{1,2,\ldots,k,(*,r)\}$. Пространство состояний $S \equiv \{X \to Z\}$. Первые k пространств эталонов $E^1$ и функционалы $\mu_1$ – те же, что и в общем случае; $E^{k+1} \equiv *$, $\mu_1(x,*) \equiv r$. Далее, если аналогичную гипотезу можно высказать относительно выборки $X^2 = X^3 \cup X^4$, то множество $(1,\ldots,(*,r) = Z$ для последующего представления выборки $X^2$ и выделения нового сложного сгустка $X^3$ строится аналогично.

Алгоритмом (k,r)-средних будем называть следующую конкретизацию изложенной конструкции: $X \subset R^J$; S(X) совпадает с множеством разбиений X на k+1 непересекающихся классов. $E(X) = E^1 \times \ldots \times E^{k+1}$, где $E^1 \equiv R^J$, $1 \le k$; $E^{k+1} \equiv *$ – точка, причем $\rho_1(x,e^1) = |x-e^1|$ для $1 \le k$, и $\rho_{k+1}(x,*) = r$. Мера несходства $\mu_1 = \rho_1$, а $M_1(e^1,s) = \sum_{x \in X^1} (\mu_1(x,e^1)^2)$. Операторы K и D определяются по формулам (2) и (3). Отметим, что при $r \to \infty$ алгоритм (k,r)-средних дает алгоритм k-средних, а при k=1 – "Форель". При этом $\mu_1$ и $M_1$ в модели алгоритма базисного эталонного типа, описывающего алгоритм "Форель", определяются так: $\mu_1(x,e^1) = |x-e^1|$, $\mu_2(x,*) \equiv r$; $M_1(e^1, s) = \sum_{x \in X^1} |x-e^1|$, $M_2(*,s) = \sum_{x \in X^2} \mu(x,*)^2 = |X^2| \cdot r^2$. Тогда получаем, что интерпретирующий функционал базисного эталонного алгоритма "Форель" – $F(s,\bar{e}) = \sum_{x \in X^1} |x - e^1|^2 + |X^2| \cdot r^2$, т.е. он имеет вид скаляризации двух функционалов, первый из которых ($F_1 = \sum_{x \in X^1} |x-e^1|^2$) оценивает разброс сгустка $X^1$ относительно эталона $e^1$, а второй ($F_2 = |X^2|$) оценивает количество не захваченных этим сгустком точек, а $r^2$ играет роль параметра скаляризации. Таким образом, становится видно, что при фиксированном параметре скаляризации r цель, описываемая функционалом, есть "минимум дисперсии в сгустке при максимуме захваченных точек"; в данном случае интерпретирующий функционал в смысле определения 3 допускает содержательную интерпретацию в качестве целевого функционала.

## §4. Адаптивный эталонный алгоритм "Пульсар"

Рассматриваемый в этом параграфе алгоритм основан на развитии идеи алгоритма "Форель". В нем не требуется априори задавать радиус r – окно просмотра выборки (т.е. в терминах §2 не конкре-

тизируется гипотеза о "достаточной удаленности" сгустка от остальной части выборки, исходя из которой задается значение r ). Задаются только границы изменения этого радиуса, и в их пределах алгоритм в процессе движения выбирает оптимальный для условий этого движения радиус ("пульсирует"). Радиус адаптируется к условиям движения, которые в алгоритме "Пульсар" определяются предысторией движения - количеством захваченных сгустком на предыдущем шаге точек, количеством произведенных к данному шагу пульсаций, текущим радиусом. Адаптация направлена на то, чтобы "удерживать", с одной стороны, величину радиуса, а с другой - количество точек в сгустке в заданных границах. Первые варианты этого алгоритма были предложены в [7].

ПРИМЕР 4. Алгоритм "Пульсар". X, P(X), S(X) и G задаются как в алгоритме (1,r)-средних из примера 3. E(X) - пространство наборов $\bar{e} = (e, r', r, m(s), \nu)$, где $e \in R^J$, r и r' - вещественные числа из отрезка $[r_{min}, r_{max}]$, а m(s) и $\nu$ - натуральные числа, причем $m(s) \in [m_{min}, m_{max}]$. $K(s,-(e,r',r,m(s),\nu)) = \{x \in X : |x-e| \le r\}$. $D((e,r',r,m(s),\nu),s) = (e_*, r_\downarrow, r_*, m_*(s), \nu_*)$, где $e_* = \frac{1}{|s|} \sum_{x \in s} x$,

$r_\downarrow = r$, $m_*(s) = |s|$, а

$$r_* = \begin{cases} \min(r+\gamma\delta,\ r_{max}), & \text{если } m_*(s) \le m_{min}, \\ \max(r-\gamma\delta,\ r_{min}), & \text{если } m_*(s) > m_{max} \text{ и при этом } \nu < \nu_{max} \text{ либо } e_* = e, \\ r - & \text{в остальных случаях.} \end{cases}$$

Здесь $\gamma = \frac{1}{1+\nu}$, а $\delta$ и $\nu_{max}$ - константы. Обратим внимание, что константа $\nu_{max}$ ограничивает движение алгоритма, только когда $e_* = e$ и одновременно $m_*(s) > m_{max}$;

$$\nu_* = \begin{cases} \nu, & \text{если } (r-r')\cdot(r_* - r_\downarrow) \ge 0, \\ \nu+1, & \text{если } (r-r')\cdot(r_* - r_\downarrow) < 0. \end{cases}$$

В качестве начальных данных выбираются произвольно или из априорных соображений подмножество $s_1 \subset X$ и набор $\bar{e}_1$.

Алгоритм "Пульсар" не описывается моделью базисного эталонного алгоритма, поскольку эталоны его состоят из элементов, часть из которых (центр сферы) являются решением экстремальной задачи, а другая часть (радиус, число точек и др.) представляет информа-

цию об истории и текущем моменте движения. Алгоритм "Пульсар" является примером алгоритма эталонного типа, в котором описание эталона дается средствами одновременно оптимизационного и структурного подходов, а движение алгоритма осуществляется на основе многокритериальной оценки достигнутого к данному шагу описания выборки [3].

## §5. Исследование сходимости эталонных алгоритмов

В этом параграфе мы приведем доказательства сходимости трех типов алгоритмов, рассмотренных в настоящей статье. В настоящее время известен ряд работ, посвященных исследованию сходимости алгоритмов эталонного типа; так, в работах [9-II] исследована сходимость алгоритма "Форель", а в работах [5] и [I2] - сходимость алгоритма k-средних. Исследование сходимости рассмотренных в настоящей статье алгоритмов может быть проведено с единых позиций, опираясь на разработанную теорию [4]. Здесь же мы приводим специфические для каждого типа алгоритмов доказательства для того, чтобы подчеркнуть ключевые моменты, обеспечивающие сходимость. Отметим, что исследование сходимости проводится нами в рамках анализа данных, т.е. без использования гипотезы о том, что выборка взята из некоторой генеральной совокупности с теми или иными свойствами. Для того чтобы не усложнять изложение техническими деталями, выборка $X$ в дальнейшем предполагается конечной.

Для класса алгоритмов, допускающих описание в терминах модели базисного эталонного алгоритма, справедлива следующая

ТЕОРЕМА I. Движение базисного эталонного алгоритма стабилизируется.

ДОКАЗАТЕЛЬСТВО. Поскольку выборка $X$ конечна, то конечно и множество состояний $S(X)$ (множество его разбиений на $k$ непересекающихся классов). Из формулы (3) следует, что дескриптор $D$ в рассматриваемой модели представляет собой отображение $D: S \to E$. Следовательно, любое движение алгоритма $(s_1,e_1),\dots,(s_n,e_n),\dots$ пробегает конечное подмножество в $S \times E$, и поэтому интерпретирующий функционал (4) принимает на движение лишь конечное число отличных друг от друга значений. Это означает, что в невозрастающей числовой последовательности $F(s_1,e_1) \geq \dots \geq F(s_n,e_n) \dots$ имеет место равенство $F(s_n,e_n) = F(s_{n+1},e_{n+1})$ для всех $n$, начиная с некоторого $n_1$. Но из этого равенства следует, что $F(s_n,e_n) = F(s_{n+1},e_n)$, что, согласно формуле (2) для оператора $K$, возможно только в случае

$s_n = s_{n+1}$. С другой стороны, $e_{n+1} = D(s_{n+1}) = D(s_n) = e_n$. Таким образом, мы показали, что $s_n = s_{n+1}$, $e_n = e_{n+1}$ для всех n, начиная с некоторого $n_1$. Что и требовалось доказать.

Перейдем теперь к исследованию сходимости эталонных алгоритмов типа "Форель". Обратим внимание, что доказательство сходимости алгоритма "Форель" вытекает из предыдущей теоремы, согласно описанию алгоритма "Форель" моделью алгоритма $(1,r)$-средних.

Пусть $\gamma$: $[0,+\infty) \to [0,1]$- невозрастающая функция, такая что $\gamma(t_0) = \lim\limits_{t \to t_0 - 0} \gamma(t)$ для всех $t_0 \in [0, +\infty]$. Для каждого натурального числа N введем новую функцию $\gamma_N(t) = \frac{[N\,\gamma(t)]}{N}$, где [ ] - символ операции взятия целой части числа. Из свойств операции [ ] следует, что $0 \le \gamma(t) - \gamma_N(t) \le \frac{1}{N}$ для всех N.

ТЕОРЕМА 2. Движение форелеподобного алгоритма с функцией принадлежности $\lambda(e,x) = \gamma_N(|x-e|)$ стабилизируется для всех N.

ДОКАЗАТЕЛЬСТВО. Для любого форелеподобного алгоритма оператор классификации K не зависит от первого аргумента, т.е. $K(s,e) \equiv K(e)$, поэтому для доказательства стабилизируемости движения такого алгоритма достаточно показать стабилизируемость последовательности описаний $e_1,\dots,e_n,\dots$. Положим $Y = \{(x,q): x \in X,\ q = 1,\dots,N\}$ и введем следующее семейство алгоритмов $A_N, N = 1,2,\dots$ классификации множества Y: S(Y) - множество всех подмножеств Y; $E(Y) \equiv R^J$; $K(s,e,[X]) = \{(x,q) \in Y,\ \gamma(|x-e|) \ge \frac{q}{N}\}$; $D(e,s) = \frac{1}{|s|} \sum\limits_{(x,q) \in s} \pi(x,q)$, где $\pi: Y \to X$, $\pi(x,q) = x$. Пусть $(s_1^N, e_1^N), \dots, (s_n^N, e_n^N), \dots$ - движение алгоритма $A_N$. Заметим, что

$$D(e_n^N, s_{n+1}^N) = \frac{1}{|s_{n+1}^N|} \sum_{x \in X} [N\gamma(|x - e_n^N|)] \cdot x .$$

Следовательно, последовательность описаний алгоритма $A_N$ совпадает с последовательностью описаний форелеподобного алгоритма с функцией принадлежности $\gamma_N(|x-e|)$. Рассмотрим функционал

$$F_N(s,e) = \sum_{(x,q)\in s} (|\pi(x,q) - e|^2 - r_q^2),$$

где $r_q = \max_t \{t: \gamma(t) \geq \frac{q}{N}\}, q = 1,\ldots,N$, и покажем, что он является интерпретирующим для алгоритма $A_N$. Пусть $(x,q) \in S^N_{n+1}$, т.е. $\delta(|x-e^N_n|) \geq \frac{q}{N}$, тогда из определения констант $r_q$ следует, что $|x-e^N_n| \leq r_q^2$. Таким образом, $F_N(s^N_{n+1}, e^N_n)$ представляет собой сумму неположительных слагаемых. В то же время, если $(x,q) \in S^N_n$ и $(|\pi(x,q) - e^N_n|^2 - r_q^2) \leq 0$, то $(x,q) \in S^N_{n+1}$, и поэтому $F_N(s^N_n, e^N_n) \geq F_N(s^N_{n+1}, e^N_n)$, причем равенство возможно тогда и только тогда, когда $s^N_n = s^N_{n+1}$. Неравенство $F_N(s^N_{n+1}, e^N_n) \geq F(s^N_{n+1}, e^N_{n+1})$ следует непосредственно из построения дескриптора D.

Применяя теперь к операторам K и D те же рассуждения, что и при доказательстве теоремы I, получаем при помощи функционала $F_N$ стабилизируемость движения алгоритма $A_N$, что влечет за собой стабилизируемость описаний рассматриваемого форелеподобного алгоритма. Доказательство теоремы закончено.

ТЕОРЕМА 3. Движение алгоритма "Пульсар" стабилизируется.

ДОКАЗАТЕЛЬСТВО. Возможны только два следующие случая.

I. Существует натуральное число $\nu_*$ такое, что $\nu_n < \nu_*$ для всех n, т.е. существует $n_*$, для которого $\nu_n = \nu_*$ для всех $n \geq n_*$. Тогда последовательность $\{r_n, n \geq n_*\}$ либо невозрастающая и ограниченная снизу, либо неубывающая и ограниченная сверху. Так как $|r_{n+1} - r_n| = \gamma_n \cdot \delta = \text{const}$ для $n \geq n_*$, то получаем, что, начиная с некоторого $n_*$, числовая последовательность $\{r_n\}$ стабилизируется. В этом случае, начиная с шага $n_*$, движение алгоритма "Пульсар" совпадает с движением алгоритма "Форель" и поэтому утверждение теоремы 3 является следствием стабилизации движения алгоритма "Форель".

2. Пусть последовательность $\nu_n$ неограниченно возрастает. Тогда существует номер $n_*$, начиная с которого $\nu_n \geq \nu_{max}$. Положим $F(s,\bar{e}) = F(s,e,r) = \sum_{x\in X} (|x-e|^2 - r^2)$. Тогда непосредственно из конструкции операторов K и D получаем $F(s_n, e_n, r_n) \geq F(s_{n+1}, e_n, r_n) =$

$= F(s_{n+1}, e_{n+1}, r_n) + |s_{n+1}| \cdot |e_n - e_{n+1}|^2 = F(s_{n+1}, e_{n+1}, r_{n+1}) +$

$+ |s_{n+1}|(|r^2_{n+1} - r^2_n| + |e_n - e_{n+1}|^2)$. Из конечной выборки X следует, что существует $\delta > 0$ такое, что если $e_n \neq e_{n+1}$, то $|e_n - e_{n+1}| > \delta$ для любого n. С другой стороны, по предположению, $\nu_n \to \infty$, и поэтому существует $n_1 > n_0$, такое, что $|r^2_{n+1} - r^2_n| < \frac{\delta^2}{2}$ для всех $n \geq n_1$. Если для некоторого $n \geq n_0$ выполняется неравенство $m_n(s) > m_{max}$, и при этом $e_n = e_{n+1}$, то $r_n = r_{n+1}$. Но в этом случае движение алгоритма, очевидно, стабилизируется. Таким образом, если движение алгоритма не стабилизируется, то либо $|e_n - e_{n+1}| \neq 0$, либо $r_{n+1} > r_n$. В каждом из этих случаев, согласно определению 3, имеет место неравенство $F(s_n, e_n, r_n) \geq F(s_{n+1}, e_{n+1}, r_{n+1})$ для всех $n \geq n_0$. Более того, как только $|e_n - e_{n+1}| \neq 0$ для $n \geq n_1$, то $F(s_n, e_n, r_n) - F(s_{n+1}, e_{n+1}, r_{n+1}) \geq \frac{\delta^2}{2} = \text{const}$, так как, очевидно, $|s_{n+1}| \geq 1$, если $|e_n - e_{n+1}| \neq 0$. Заметим теперь, что из предположения $\nu_n \to \infty$ и формулы для $r_n$ следует, что $|e_n - e_{n+1}| \neq 0$ для бесконечного числа номеров. Т.е. неравенство $F(s_n, \bar{e}_n) - F(s_{n+1}, \bar{e}_{n+1}) \geq \text{const}$ должно выполняться для бесконечной подпоследовательности номеров, что противоречит ограниченности функционала F. Следовательно, последовательность $\nu_n$ не может быть неограниченной.

Таким образом, при любом движении алгоритма "Пульсар" последовательность $\nu_n$ ограничена сверху (но необязательно константой $\nu_{max}$), а из этого, как показано выше, следует стабилизация движения алгоритма.

## Заключение

Изложенные в настоящей статье методы построения алгоритмов классификации опираются на следующие основные положения. Многообразие всех алгоритмов автоматической классификации можно описать при помощи иерархической структуры. На самом верхнем уровне находится достаточно универсальная математическая модель, составляющие элементы которой образуют средство для единообразного описания алгоритмов автоматической классификации (§1). На самом нижнем уровне располагаются движения конкретных алгоритмов. Переход с высших уровней на низшие происходит за счет конкретизаций, наполняющих элементы структуры алгоритмов информацией о характере данных, конечной цели классификации, априорных гипотезах о расположении выборки в пространстве наблюдений. Предложены методы целенаправленного конструирования алгоритмов, основанные на переходах снизу-вверх и сверху-вниз по уровням такой иерархии. На этом пути уда-

ется получать семейства алгоритмов, вводя различные модификации априорных сведений и гипотез, конкретизация которых порождает различные исходные алгоритмы.

Введение модификаций гипотезы, обосновывающей применимость алгоритма "Форель", позволила построить семейство форелеподобных алгоритмов (§2). В §3 за счет модификации конечной цели классификации осуществлен "подъем" известного алгоритма k-средних, $k \geq 2$, до алгоритма $(k,r)$-средних, $k \geq 1, r \leq \infty$, что дало возможность включить алгоритм "Форель", как алгоритм $(1,r)$-средних, в новое двупараметрическое семейство алгоритмов. Важный пример перехода от движения с фиксированным "окном" просмотра выборки ("Форель") к движению с адаптивным восстановлением оптимального окна дан в §4. Идея этого перехода, задающая семейство адаптивных алгоритмов, состоит в том, что задание управляющих параметров алгоритма заменяется заданием способа восстановления оптимальных параметров, адаптированных к условиям движения алгоритма.

## Литература


1. АЙВАЗЯН С.А., БЕЖАЕВА З.И., СТАРОВЕРОВ О.В. Классификация многомерных наблюдений. -М.: Статистика, 1974.

2. АЙВАЗЯН С.А., БУХШТАБЕР В.М. Первая Всесоюзная школа-семинар "Программно-алгоритмическое обеспечение прикладного многомерного статистического анализа". -Успехи мат.наук, 1980, т. 35, вып. 4.

3. БУХШТАБЕР В.М., МАСЛОВ В.К. Задачи прикладной статистики как экстремальные задачи на нестандартных областях. -В кн.: Алгоритмическое и программное обеспечение прикладного статистического анализа. М., 1980.

4. БУХШТАБЕР В.М., МАСЛОВ В.К., ЗЕЛЕНЮК Е.А. Модели и алгоритмы автоматической классификации данных (систематизация и целенаправленное конструирование). -В кн.: Тезисы докладов 2-й всесоюзной научно-технической конференции "Применение многомерного статистического анализа в экономике и оценке качества продукции". Тарту, 1981.

5. DIDAY E. et collaborateurs. Optimisation en classification automatique.-I.N.R.I.A., Paris,1980.

6. ЗАГОРУЙКО Н.Г. Методы распознавания и их применение. -М.: Сов. радио, 1972.

7. МАСЛОВ В.К. Исследование непараметрических методов поиска информативных признаков и построения решающих правил в задачах распознавания. Автореф. дис. -М., 1973. (ИППИ АН СССР).

8. МЕШАЛКИН Л.Д. Параметризация многомерных распределений.-В кн.: Прикладной многомерный статистический анализ. М., 1978.

9. БЛЕХЕР П.М., КЕЛЬБЕРТ М.Я. Доказательство сходимости алгоритма "Форель". -В кн.: Прикладной многомерный статистический анализ. М., 1978.

I0. ОРЛОВ А.И. Сходимость эталонных алгоритмов. -В кн.: Прикладной многомерный статистический анализ. М., 1978.

II. СЕВРЮК М.Б. Сходимость алгоритма "Форель" для бесконечного числа объектов. -В кн.: Алгоритмическое и программное обеспечение прикладного статистического анализа. М., 1980.

I2. Классификация и кластер /Под ред. Дж.Вэн Райзина. - М.: Мир, 1981.

Поступила в ред.-изд.отд.
31 марта 1981 года