ЭКСТРЕМАЛЬНЫЕ ЗАДАЧИ, ВОЗНИКАЮЩИЕ ПРИ ОБОСНОВАНИИ ЭВРИСТИЧЕСКИХ ПРОЦЕДУР

Ю. И. Журавлев

Москва

Существует значительное количество методов (особенно в задачах построения экстремумов функций многих переменных, задачах распознавания образов и т. д.), которые, не имея строгого обоснования, дают значительный эффект при решении практических задач. Такие методы обычно формируются при наблюдении за исследуемыми объектами или получаются из «правдоподобных» соображений. После разработки методы проверяются на некотором количестве эталонных задач. Если эффективность их решения оказывается удовлетворительной, методы получают право на существование.

Процедуры указанного вида обычно называют эвристическими. В большинстве случаев для эвристических алгоритмов нетрудно указать задачу, при решении которой алгоритм не только не эффективен, но дает заведомо неправильное решение. Однако их авторы указывают, что подобные примеры не приводят к компрометации эвристических методов, так как при решении прикладных задач специального вида алгоритмы хорошо работают.

Появление и широкое распространение эвристических процедур связано прежде всего с тем, что существует большое количество прикладных задач, при решении которых трудно или невозможно применить методы, обоснованные по всем стандартам математической строгости. По-видимому, следует считать широкое распространение эвристических процедур совершившимся фактом и попытаться построить строгую теорию «нестрогих» процедур и методов.

Здесь возникает множество проблем: когда следует считать эвристический алгоритм обоснованным настолько, что им можно пользоваться при решении прикладных задач, как сравнивать между собой такие алгоритмы, как проводить хотя бы частичное обоснование и т. п.

В настоящей работе мы рассмотрим один подход к проблеме обоснования эвристических алгоритмов, их сравнения и построения наилучших, в некотором смысле, нестрогих алгоритмов.

§ 1. Основные понятия. Постановка задачи

Рассмотрим три множества: $\{A\}$, $\{Z\}$, $\{\tilde{Z}\}$, $\{\tilde{Z}\} \subseteq \{Z\}$. Элементы A множества $\{A\}$ назовем алгоритмами, элементы множества $\{Z\}$ — задачами и элементы $\tilde{Z}$ множества $\{\tilde{Z}\}$ — эталонными задачами, или эталонами.

Будем считать, что на множестве $\{Z\}$ для каждого алгоритма A из $\{A\}$ существует функция $\varphi_A(Z)$ — эффективность решения задачи Z алгоритмом A, $\varphi_A(Z) \geqslant 0$. Например:

1. Если $\{Z\}$ — совокупность задач нахождения абсолютного экстремума функций $f(x_1, x_2, \ldots, x_n)$ из заданного класса, $\{A\}$ — набор алгоритмов нахождения абсолютного экстремума таких функций, m_f — абсолютный

экстремум функции f, $m_f(A)$ — результат применения к f алгоритма A, то в качестве функции эффективности можно рассматривать

$$\varphi_A(f) = \frac{1}{|m_f - m_f(A)| + 1}.$$

2. Если $\{Z\}$ есть совокупность задач вычисления фиксированного предиката P на различных элементах M множества $\mathfrak{M}$, $\{A\}$ — совокупность алгоритмов для решения этих задач, $P(M)$ — значение P на M, $P_A(M)$ — значение предиката P на M, вычисленное при помощи алгоритма A, то можно положить:

$$\varphi_A(M) = \begin{cases} 1, & \text{если } P(M) = P_A(M), \\ 0, & \text{если } P(M) \neq P_A(M). \end{cases}$$

Будем также считать, что для эталонных задач $\tilde{Z}$ функция $\varphi_A(\hat{Z})$ задана или может быть просто вычислена ($A \in \{A\}$).

Пусть $[\varphi_A(\tilde{Z}), \tilde{Z}]$ — совокупность всех пар, таких, что $\tilde{Z} \in \{\tilde{Z}\}$.

Будем считать, что на множестве эталонных задач задана эффективность ψ_A алгоритма A, $A \in \{A\}$, при условиях:

1⁰. если $\psi_A = \psi_A[\varphi_A(\tilde{Z}), \tilde{Z}] \geqslant 0$,

2⁰. если для любой задачи $\tilde{Z}$ выполнено неравенство

$$\varphi_A(\tilde{Z}) \leqslant \varphi_B(\tilde{Z}), \quad A, \ B \in \{A\}, \text{ то } \psi_A \leqslant \psi_B.$$

Рассмотрим примеры функций ψ_A:

1) $\psi_A = \inf\limits_{\tilde{Z} \in \{\tilde{Z}\}} \varphi_A(\tilde{Z})$;

2) $\psi_A = \dfrac{1}{\mu\{\tilde{Z}\}} \displaystyle\int\limits_{\{\tilde{Z}\}} \varphi_A(\tilde{Z})\, d\mu\{\tilde{Z}\}$,

где $\{\tilde{Z}\}$ — измеримое множество и $\mu\{\tilde{Z}\}$ — мера на $\{\tilde{Z}\}$;

3) $\psi_A = \dfrac{1}{\mu\{\tilde{Z}\}} \displaystyle\int\limits_{\{\tilde{Z}\}} \varphi_A(\tilde{Z})\, V(\tilde{Z})\, d\mu\{\tilde{Z}\}$,

где $V(Z) \geqslant 0$ — важность задачи Z, определенная на множестве $\{Z\}$.

Очевидно, функции ψ_A, определенные соотношениями 1)—3), удовлетворяют условиям 1⁰—2⁰.

Будем считать, что в множестве алгоритмов $\{A\}$ существует хотя бы один алгоритм A^*, на котором реализуется $\max \psi_A$. Такие алгоритмы A^* в дальнейшем будем называть экстремальными.

Очевидно, что вместо функций эффективности $\varphi_A(Z)$, ψ_A можно рассматривать функции неэффективности $\tilde{\varphi}_A(Z)$, $\tilde{\psi}_A$. В этом случае естественно считать экстремальными алгоритмы, на которых реализуется $\min\limits_{A \in \{A\}} \tilde{\psi}_A$.

В дальнейшем мы будем рассматривать экстремальные алгоритмы и способы их построения. Заметим, что если экстремальный алгоритм найден и его применение к решению задач $\{Z\}$ достаточно эффективно, то, в некотором смысле, решена задача об обосновании алгоритмов из $\{A\}$ и их сравнении между собой.

§ 2. Задачи распознавания образов

Пусть заданы метрические пространства $M_1, M_2, \ldots, M_n$ с метриками $\rho_1, \rho_2, \ldots, \rho_n$ соответственно.

Назовем набор $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ допустимым, если $\alpha_i \in M_i$, $i = 1, 2, \ldots, n$. Обозначим множество всех допустимых наборов через D. В дальнейшем будем предполагать, что существует разбиение D на систему непересекающихся подмножеств $K_1, K_2, \ldots, K_l$, которые будем называть классами.

Пусть также заданы множества M_R и M_K такие, что:

1°. $M_R \subseteq D$, $\quad M_K \subseteq D$;

2°. $K_i \cap M_R$ непусто, $i = 1, 2, \ldots, l$.

Множество M_R назовем решающим и M_K — контрольным, если известны (заданы) $K_i \cap M_R$, $K_i \cap M_K$, $i = 1, 2, \ldots, l$.

Определение 1. Алгоритм A решает задачу распознавания по решающему множеству M_R, если для любого элемента D с использованием только информации об элементах M_R алгоритм A выдает номер из множества $\{0, 1, 2, \ldots, l\}$.

Если алгоритм выдал номер i, отличный от 0, то будем считать, что A занес предъявленный элемент из D в класс K_i. При выдаче символа 0 алгоритм отказался от классификации (распознавания) элемента из D.

В дальнейшем будем считать множества M_R, M_K, K_1, K_2, $\ldots$, K_l измеримыми. Положим $M_K \cap K_i = \tilde{K}_i$, $i = 1, 2, \ldots, l$.

Сопоставим алгоритму A разбиение множеств $\tilde{K}_i$ на непересекающиеся подмножества $\tilde{K}_{ij}$, $j = 0, 1, 2, \ldots, l$. В множество $\tilde{K}_{ij}$ отнесем все объекты из $\tilde{K}_i$, которые алгоритм A зачислил в класс K_j, $j \neq 0$. В множество $\tilde{K}_{i0}$ отнесем все элементы из $\tilde{K}_i$, которые алгоритм A отказался классифицировать.

Определение 2. Величину

$$\psi_A = \frac{1}{\mu(M_K)} \sum_{i=1}^{l} \mu(\tilde{K}_{ii}) \tag{1}$$

назовем эффективностью алгоритма A, если все $\tilde{K}_{ii}$, $i = 1, 2, \ldots, l$ измеримы. Здесь $\mu(M_K)$, $\mu(\tilde{K}_{ii})$, $i = 1, 2, \ldots, l$, — меры множеств M_K, $\tilde{K}_{ii}$ соответственно.

В реальных задачах классы $K_1, \ldots, K_l$ не всегда важны в равной степени. Может оказаться, что правильное распознавание объектов из K_i более важно, чем объектов из K_j.

Если заданы величины $\gamma_1, \gamma_2, \ldots, \gamma_l$, характеризующие «важность» классов $K_1, \ldots, K_l$, то эффективностью алгоритма A естественно считать величину ψ_A, которая задается следующим равенством:

$$\psi_A = \frac{1}{\mu(M_K)} \sum_{i=1}^{l} \gamma_i \mu(\tilde{K}_{ii}). \tag{2}$$

Пусть задана совокупность $\{A\}$ алгоритмов, решающих задачу распознавания.

Определение 3. Алгоритм A назовем экстремальным алгоритмом первого рода, если на нем реализуется

$$\sup_{A \in \{A\}} \psi_A.$$

Здесь ψ_A может быть задана равенствами (1) или (2).

Введем в рассмотрение величины α_{ij}, $i = 1, 2, \ldots, l$, $j = 1, 2, \ldots, l$, $i \neq j$, и величины β_i, $i = 1, 2, \ldots, l$. Будем называть α_{ij} штрафом за отнесение элемента из класса K_i в класс K_j и β_i — штрафом за отказ от распознавания элемента из K_i алгоритмом распознавания.

Определение 4. Величину

$$\tilde{\psi}_A = \frac{1}{\mu(M_K)} \left(\sum_{i,j} \alpha_{ij} \mu(\tilde{K}_{ij}) + \sum_{i} \beta_i \mu(\tilde{K}_{i0}) \right) \tag{3}$$

$$i = 1, 2, \ldots, l; \; j = 1, 2, \ldots, i-1, i+1, \ldots, l$$

назовем неэффективностью алгоритма A, если все множества $\tilde{K}_{ij}$, $1 \leqslant i \leqslant l$; $1 \leqslant j \leqslant l$, измеримы. Здесь $\mu(\tilde{K}_{ij})$ — мера множества $\tilde{K}_{ij}$.

Заметим, что в случае, когда классы $K_1, \ldots, K_l$ равноценны, естественно считать все α_{ij} равными между собой.

Определение 5. Алгоритм A называется экстремальным алгоритмом второго рода, если на нем достигается нижняя грань $\tilde{\varphi}_A$.

Наконец, если заданы также величины α_{ii}, $i = 1, 2, \ldots, l$, — меры поощрения за правильное распознавание элементов класса K_i, естественно определять эффективность ψ_A следующим образом:

$$\psi_A = \left| \sum_i \alpha_{ii}\mu(\tilde{K}_{ii}) - \left(\sum_{i,j} \alpha_{ij}\mu(\tilde{K}_{ij}) + \sum_i \beta_i\mu(\tilde{K}_{i0})\right) \right| \frac{1}{\mu(M_k)}, \tag{4}$$
$$1 \leqslant i \leqslant l;\ 1 \leqslant j \leqslant l;\ i \neq j.$$

Алгоритм A, на котором достигается верхняя грань ψ_A (см. (4)), назовем экстремальным алгоритмом третьего рода.

Рассмотрим процедуры отыскания экстремальных алгоритмов. Дальнейшее изложение распадается на следующие этапы.

I. Описание множеств D, M_R, $M_{\tilde{K}}$.

II. Описание множества $\{A\}$.

III. Описание способа вычисления φ_A, $\tilde{\varphi}_A$.

IV. Описание оптимизации в пространстве алгоритмов.

I. Будем рассматривать конечные множества M_R, M_K допустимых наборов $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha_i \in M_i$, $i = 1, 2, \ldots, n$, и задавать их в виде таблиц. В таблицах наборы по вертикали упорядочены следующим образом. Сначала выписываются элементы класса K_1, затем элементы класса K_2 и т. д. Объекты задаются строками таблиц и обозначаются символами S_i, S'_i. В дальнейшем примем, что множество M_R состоит из объектов $S_1, S_2, \ldots, S_m$, причем к первому классу K_1 относятся объекты $S_1, S_2, \ldots, S_{m_1}$, классу K_2 принадлежат объекты $S_{m_1+1}, \ldots, S_{m_2}, \ldots$, классу K_j — объекты $S_{m_{j-1}+1}, \ldots, S_{m_j}, \ldots$, классу K_l — объекты $S_{m_{l-1}+1} \ldots, S_{m_l}$. Аналогично, M_K положим составленным из объектов $S'_1, \ldots, S'_q$, причем объекты $S'_{q_{j-1}+1}, \ldots, S'_{q_j}$ принадлежат классу K_j, $j = 1, 2, \ldots, l$, $q_l = q$.

Нам требуется теперь описать класс алгоритмов $\{A\}$, которые по таблице M_R и предъявленному допустимому объекту S вырабатывают один из номеров $\{0, 1, 2, \ldots, l\}$. Рассмотрим в настоящей работе один класс $\{A\}$. Аналогичные построения могут быть проведены и для других множеств $\{A\}$. Существенно здесь то, что класс $\{A\}$ должен допускать параметризацию. Это означает, что алгоритмы из $\{A\}$ могут быть взаимно однозначно закодированы наборами значений фиксированных числовых параметров $x_1, x_2, \ldots, x_t$.

II. Рассмотрим множество $\{1, 2, \ldots, n\}$ и обозначим через $\{\Omega_k\}$ совокупность всех его подмножеств мощности k, $1 \leqslant k \leqslant n$.

Пусть $\Omega_k \in \{\Omega_k\}$ и $\Omega_k = \{r_1, r_1, \ldots, r_k\}$.

Выделим в таблице, задающей множество M_R, и строке, задающей объект S, координаты с номерами $r_1, r_2, \ldots, r_k$ и удалим все остальные столбцы в M_R и координаты в S. Полученные после указанных операций строки обозначим через $S_i(\Omega_k)$, $S(\Omega_k)$, $i = 1, 2, \ldots, m$.

Рассмотрим величины $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, ε, $\varepsilon_i \geqslant 0$, $\varepsilon \geqslant 0$, $\varepsilon < k$. Пусть $S_i(\Omega_k) = (\alpha_{r_1}, \alpha_{r_2}, \ldots, \alpha_{r_k})$, $S(\Omega_k) = (\beta_{r_1}, \beta_{r_2}, \ldots, \beta_{r_k})$. Напишем неравенства: $\rho_{r_1}(\alpha_{r_1}, \beta_{r_1}) \leqslant \varepsilon_{r_1}, \ldots, \rho_{r_k}(\alpha_{r_k}, \beta_{r_k}) \leqslant \varepsilon_{r_k}$. Обозначим число невыполненных неравенств через δ.

Определение 6. Строки S и S_i назовем близкими по Ω_k, если $\delta \leqslant \varepsilon$.

Пусть $\{S_i\}^j$ — множество всех строк из таблицы M_R, близких к S по множеству Ω_k, $j = 1, 2, \ldots, l$; $\{S_i\}^j \subseteq K_j$.

Рассмотрим неотрицательные величины $\gamma(S_1), \ldots, \gamma(S_m)$, сопоставленные строкам таблицы M_R. Положим

$$\Gamma^j_{\Omega_k}(S) = \sum_{S_i \in \{S_i\}_j} \gamma(S_i). \tag{5}$$

Далее обозначим

$$\Gamma_j(S) = \sum_{\Omega_k \in \{\Omega_k\}} \Gamma^j_{\Omega_k}(S), \quad j = 1, 2, \ldots, l. \tag{6}$$

Величины $\Gamma_j(S)$ указывают на близость строки S к элементам класса K_j, входящим в таблицу M_R. Очевидно, $\Gamma_j(S)$ являются функциями M_R, S, а также параметров k, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, ε, $\gamma(S_1) = \gamma_1$, $\gamma(S_2) = \gamma_2, \ldots, \gamma(S_m) = \gamma_m$. Варьируя эти параметры, мы сможем получить различные алгоритмы для построения величин $\Gamma_1(S), \ldots, \Gamma_l(S)$.

Алгоритм A заносит S в класс K_j, если:

1^0. $\Gamma_j(S) - \Gamma_i(S) \geqslant \Delta_1$;

2^0. $\Gamma_j(S) \Big/ \sum_{i=1}^{l} \Gamma_i(s) \geqslant \Delta_2$, Δ_1, Δ_2 — заданные величины, $1 \leqslant j \leqslant l$, $j \neq i$, $\Delta_1 > 0$, $\Delta_2 > 0$.

В случае, когда ни для одного номера j не выполняются одновременно неравенства 1^0, 2^0, алгоритм A отказывается от классификации S.

Мы описали множество $\{A\}$ алгоритмов распознавания. Алгоритмы этого класса кодируются набором значений параметров k, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, ε, $\gamma_1, \gamma_2, \ldots, \gamma_m$, Δ_1, Δ_2; всего $n + m + 4$ параметра. Для алгоритмов описанного класса функционалы ψ_A, $\tilde{\psi}_A$ становятся функциями числовых аргументов

$$\psi_A = \psi_A(k, \varepsilon_1, \ldots, \varepsilon_n, \varepsilon, \gamma_1, \ldots, \gamma_m, \Delta_1, \Delta_2),$$

$$\tilde{\psi}_A = \tilde{\psi}_A(k, \varepsilon_1, \ldots, \varepsilon_n, \varepsilon, \gamma_1, \ldots, \gamma_m, \Delta_1, \Delta_2).$$

Задача построения экстремального распознающего алгоритма сводится, таким образом, к отысканию верхней или нижней грани функции многих переменных.

Заметим, что можно ограничиться следующими значениями введенных параметров:

$$1 \leqslant k \leqslant n-1; \; 0 \leqslant \varepsilon \leqslant k-1; \; 0 \leqslant \gamma_i \leqslant 1, \; i = 1, 2, \ldots, n; \; 0 < \Delta_2 \leqslant 1.$$

Если множества $M_1, M_2, \ldots, M_n$ ограниченны и $\rho_i^{\max}$ — наибольшее расстояние между элементами M_i, то параметр ε_i достаточно варьировать в отрезке $[0, \rho_i^{\max}]$. Параметр Δ_1 можно ограничить сверху величиной $\left(\sum_{i=1}^{m} \gamma_i\right) C_n^k$.

III. Выведем формулу для эффективного вычисления величин $\Gamma_j(S)$. Пусть $S_i = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $S = (\beta_1, \beta_2, \ldots, \beta_n)$. Количество выполненных неравенств из совокупности $\rho_1(\alpha_1, \beta_1) \leqslant \varepsilon_1, \ldots, \rho_n(\alpha_n, \beta_n) \leqslant \varepsilon_n$ обозначим через $r(S, S_i)$.

Теорема 1 (см. [1]).

$$\Gamma_j(S) = \sum_{i=m_{j-1}+1}^{m_j} \gamma(S_i) \sum_{t=0}^{\varepsilon} C^{k-t}_{r(S, S_i)} C^t_{n-r(S, S_i)}.$$

Доказательство. Рассмотрим строку S_i из класса K_j и подсчитаем число наборов множества $\{\Omega_k\}$, по которым эта строка близка к строке S (определение 6). Последнее имеет место, если среди неравенств $\rho_{i_1}(\alpha_{i_1}, \beta_{i_1}) \leqslant \varepsilon_{i_1}, \ldots, \rho_{i_k}(\alpha_{i_k}, \beta_{i_k}) \leqslant \varepsilon_{i_k}$ не выполняется не более ε.

Заметим, что все k неравенств будут выполняться для $C^k_{r(S, S_i)}$ наборов Ω_k. Одно неравенство не выполняется и остальные $k-1$ выполняются для $C_{r(\bar{S},\, S_i)}^{k-1} C_{n-r(S,\, S_i)}^{1}$ наборов Ω_k, и т. д.; t неравенств не выполняются и $k-1$ выполняется для $C_{r(\bar{S},\, S_i)}^{k-t} C_{n-r(S,\, S_i)}^{t}$ наборов Ω_k.

Поэтому общее число наборов Ω_k, по которым близки строки S и S_i, выражается следующей формулой:

$$\sum_{t=0}^{\varepsilon} C_{r(\bar{S},\, S_i)}^{k-t} C_{n-r(S,\, S_i)}^{t} = A. \tag{7}$$

Подставив (5) в (6) и переменив порядок суммирования, получим:

$$\Gamma_j(S) = \sum_{i=m_j+1}^{m_j} \gamma(S_i) A.$$

Последнее равенство доказывает теорему.

Заметим, что если существует эффективный алгоритм для вычисления расстояний $\rho_i(\alpha_i, \beta_i)$ и число операций при одном таком вычислении не превосходит Q, то число операций при вычислении всех величин $\Gamma_1(S)$, $\Gamma_2(S), \ldots, \Gamma_l(S)$ не превосходит $2Qnm$. Число операций при распознавании одного объекта в фиксированном алгоритме A пропорционально «площади» таблицы M_R с коэффициентом пропорциональности, не превосходящим $2Q$.

Можно рассматривать различные подклассы в множестве алгоритмов $\{A\}$, например:

1) $\gamma_1 = \gamma_2 = \ldots = \gamma_m = 1$; все элементы M_R равноценны; $\varepsilon_1 = \varepsilon_2 = \ldots = \varepsilon_n = \varepsilon = 0$. Тогда получается семейство алгоритмов, зависящее от трех параметров k, Δ_1, Δ_2, — семейство $\{A\}_3$. В указанном случае

$$\Gamma_j(S) = \sum_{i=m_j+1}^{m_j} C_{r(S,\, S_i)}^{k}, \quad j = 1, 2, \ldots, l;$$

2) $\gamma_1 = \gamma_2 = \ldots = \gamma_m = 1$, $\varepsilon_1 = \varepsilon_2 = \ldots = \varepsilon_n = 0$. В этом случае семейство алгоритмов зависит от четырех параметров k, ε, Δ_1, Δ_2.

Для рассматриваемого нами случая конечных множеств M_R, M_K будем считать, что мера множества совпадает с его мощностью. Тогда для вычисления величин ψ_A, $\tilde{\psi}_A$ достаточно провести процесс распознавания последовательно для строк $S_1', \ldots, S_q'$ таблицы M_K. После этого нетрудно вычислить мощности множеств $\tilde{K}_{ij}$, $i = 1, 2, \ldots, l$; $j = 0, 1, 2, \ldots, l$, а следовательно, и величины ψ_A, $\tilde{\psi}_A$ для фиксированного алгоритма A.

IV. Рассмотрим вопросы, связанные с отысканием экстремальных алгоритмов первого, второго и третьего рода. Мы считаем, что мощность множества совпадает с его мерой и пространства $M_1, M_2, \ldots, M_n$ ограниченны. Области изменения всех параметров, кроме Δ_1, Δ_2, замкнуты. Значения ψ_A, $\tilde{\psi}_A$ при фиксированных M_R и M_K ограниченны.

Изучим влияние параметров Δ_1, Δ_2 на величины ψ_A, $\tilde{\psi}_A$. Рассмотрим произвольную строку S_i' из таблицы M_K. Задавая параметры K, ε, $\varepsilon_2, \ldots, \varepsilon_n$, ε, $\gamma_1, \ldots, \gamma_m$, вычислим величины $\Gamma_1(S_i'), \ldots, \Gamma_l(S_i')$. Они не зависят от Δ_1, Δ_2. Возможны следующие случаи:

а) в наборе $\Gamma_1(S_i'), \ldots, \Gamma_l(S_i')$ существуют, по крайней мере, два равных по величине элемента. Тогда при любом Δ_1, отличном от нуля, алгоритм откажется распознавать строку S_i';

б) в наборе $\Gamma_1(S_i'), \ldots, \Gamma_l(S_i')$ существует единственный максимальный элемент $\Gamma_j(S_i')$. Положим

$$\frac{\Gamma_j(S_i')}{\sum_{j=1}^{l} \Gamma_j(S_i')} = \alpha(S_i'), \quad \min_{\substack{t \neq j \\ 1 \leqslant t \leqslant l}} (\Gamma_j(S_i') - \Gamma_t(S_i')) = \beta(S_i').$$

Заметим, что если не выполнено хотя бы одно из неравенств

$$\Delta_1 \leqslant \alpha(S'_i), \quad \Delta_2 \leqslant \beta(S'_i),$$

то алгоритм A откажется от распознавания строки S'_i. При одновременном выполнении указанных неравенств алгоритм A некоторым способом классифицирует строку S'_i. Пусть

$$\mathfrak{A} = \min_{S'_i \in M_K} \alpha(S'_i), \quad \mathfrak{B} = \min_{S'_i \in M_K} \beta(S'_i).$$

Легко видеть, что при фиксированных параметрах $\gamma_1, \ldots, \gamma_m, \varepsilon_1, \ldots$ $\ldots, \varepsilon_n$ —$\sup \psi_A$, $\inf \tilde{\psi}_A$ совпадают с $\max \psi_A$, $\min \tilde{\psi}_A$ соответственно, если $1 \leqslant k \leqslant n-1$, $1 \leqslant \varepsilon \leqslant k-1$, ε, k — целые, $\mathfrak{A} \leqslant \Delta_1 \leqslant C_n^k \left(\sum_{i=1}^m \gamma_i\right)$, $\mathfrak{B} \leqslant \Delta_2 \leqslant 1$. Нами доказана

Теорема 2. *Для семейств алгоритмов, зависящих от параметров k, ε, Δ_1, Δ_2, экстремальные алгоритмы первого, второго и третьего рода кодируются наборами параметров из замкнутой области:* $1 \leqslant k \leqslant n-1$; $1 \leqslant \varepsilon \leqslant k-1$; $\mathfrak{A} \leqslant \Delta_1 \leqslant C_n^k \left(\sum_{i=1}^m \gamma_i\right)$, $\mathfrak{B} \leqslant \Delta_2 \leqslant 1$.

Обозначим через T множество значений параметров k, $\varepsilon_1, \ldots, \varepsilon_n$, ε, $\gamma_1, \ldots, \gamma_m$, Δ_1, Δ_2, определяемое условиями: k, ε — целые числа, $\varepsilon \leqslant k$, $1 \leqslant k \leqslant n-1$, $0 \leqslant \varepsilon_i \leqslant \rho_i^{\max}$, $i = 1, 2, \ldots, n$, $0 \leqslant \gamma_i \leqslant 1$, $i = 1, 2, \ldots, m$, $0 < \Delta_1 \leqslant C_n^k \left(\sum_{i=1}^m \gamma_i\right)$, $0 < \Delta_2 \leqslant 1$.

Теорема 3. $\operatorname{Inf}_T \tilde{\psi}_A$, $\sup_T \psi_A$ *достигаются в точках множества* T.

Доказательство. Каждая строка S'_i из M_K любым алгоритмом A может быть либо отнесена к одному из l классов, либо не классифицирована. Таким образом, число различных состояний S'_i в алгоритме A конечно. Число строк в M_K также конечно. Поэтому функционалы ψ_A, $\tilde{\psi}_A$, определенные в (2)—(4) (они же являются функциями, заданными на T), принимают на заданном пространстве $\{A\}$ (на T) лишь конечное число значений. Из последнего факта легко следует доказательство теоремы.

Теоремы 2 и 3 показывают, что задача построения экстремальных алгоритмов полностью сводится к отысканию максимумов или минимумов функции многих переменных. Для отыскания экстремумов могут быть применены методы переборного типа (при небольшом числе целочисленных параметров), методы градиентного типа или методы случайного поиска.

Следует отметить, что даже при больших таблицах M_R и M_K задача построения локального (а иногда и глобального) экстремума эффективно решается с применением современных ЭВМ. Это связано с тем, что вид оптимизируемых функций весьма специфичен. Использование указанной специфики дает возможность строить достаточно экономичные алгоритмы оптимизации.

§ 3. Веса объектов в системе

Пусть снова заданы множества $\{\tilde{Z}\}$, $\{Z\}$ и набор алгоритмов $\{A\}$. Будем считать, что все алгоритмы из $\{A\}$ используют лишь строго определенную информацию о задачах из $\{Z\}$. Заранее задано множество x_1, $x_2, \ldots, x_n$ параметров, которые полностью определяют задачу Z. Алгоритмы из $\{A\}$ решают задачу Z, используя все (или только некоторые) значения параметров $x_1, x_2, \ldots, x_n$. Алгоритмы используют также информацию о ранее решенных задачах. Здесь подразумевается, что имеется таблица, в строках которой записаны значения параметров ранее решенных задач и полученные ответы.

Предположим сначала, что решение проводится при наличии полной информации о задаче с использованием алгоритма A. Это означает, что известны значения всех параметров, описывающих решаемую задачу Z, а также значения всех параметров ранее решенных задач. Пусть при этих условиях эффективность алгоритма A есть ψ_A.

Удалим теперь из описания задач Z значения параметров $x_{i_1}, \ldots, x_{i_k}$. В новых условиях вычислим эффективность алгоритма A. Пусть она равна $\psi_A(x_{i_1}, \ldots, x_{i_k})$.

Определение 7. Величину $\psi_A - \psi_A(x_{i_1}, \ldots, x_{i_k}) = P(x_{i_1}, \ldots, x_{i_k})$ назовем весом или важностью набора параметров $x_{i_1}, \ldots, x_{i_k}$ в алгоритме A. Если $k=1$, величину $\psi_A - \psi_A(x_{i_1})$ назовем важностью параметра x_{i_1} в алгоритме A.

Пусть $A = A^*$ есть экстремальный алгоритм (§ 1).

Определение 8. Величину $\psi_{A^*} - \psi_{A^*}(x_{i_1}, \ldots, x_{i_k})$ назовем весом $P^*(x_{i_1}, \ldots x_{i_k})$ набора параметров $x_{i_1}, \ldots, x_{i_k}$.

Понятие веса набора параметров может быть несколько видоизменено. Пусть при кодировании задачи набором признаков $x_1, x_2, \ldots, x_n$ экстремальным является алгоритм A^*. После удаления из описания задачи параметров $x_{i_1}, \ldots, x_{i_k}$ задачи Z, вообще говоря, переходят в другие задачи $Z_{x_{i_1}, \ldots, x_{i_k}}$.

Находим экстремальный в том же смысле алгоритм, используя вместо множества $\{\tilde{Z}\}$ множество $\{\tilde{Z}_{x_{i_1}, \ldots, x_{i_k}}\}$. Обозначим новый экстремальный алгоритм через $\tilde{A}^*$. Пусть $\psi_{A^*}, \psi_{\tilde{A}^*}$ — эффективности алгоритмов A^*, $\tilde{A}^*$ соответственно.

Определение 9. Величину $\tilde{P}^*(x_{i_1}, \ldots, x_{i_k}) = \psi_{A^*} - \psi_{\tilde{A}^*}$ назовем экстремальным весом набора параметров $x_{i_1}, \ldots, x_{i_k}$.

Рассмотрим применение введенных понятий к задачам распознавания образов [2]. Пусть по множествам M_R и M_K построен экстремальный алгоритм A^* и ψ_{A^*} — эффективность A^*.

Удалим из M_R и M_K столбцы с номерами $i_1, i_2, \ldots, i_t$. Полученные таблицы обозначим через $M_R(i_1, i_2, \ldots, i_t)$, $M_K(i_1, i_2, \ldots, i_t)$. По этим таблицам вычислим снова эффективность алгоритма A^* — величину $\psi_A(i_1, i_2, \ldots, i_t)$. Очевидно, $\psi_{A^*} - \psi_{A^*}(i_1, i_2, \ldots, i_t)$ есть вес набора параметров (признаков) с номерами $i_1, i_2, \ldots, i_t$.

Построим по таблицам $M_R(i_1, i_2, \ldots, i_t)$, $M_K(i_1, i_2, \ldots, i_t)$ новый экстремальный алгоритм $\tilde{A}^*$ и вычислим его эффективность $\psi_{\tilde{A}^*}(i_1, i_2, \ldots, i_t)$. Очевидно, $\psi_{A^*} - \psi_{\tilde{A}^*}(i_1, i_2, \ldots, i_t)$ есть экстремальный вес набора параметров (признаков) с номерами $i_1, \ldots, i_t$.

ЛИТЕРАТУРА

1. *Ю. И. Журавлев, В. В. Никифоров.* Алгоритмы распознавания, основанные на вычислении оценок. — Кибернетика, 1971, № 3.
2. *Ю. И. Журавлев, Ш. Туляганов.* Информационный вес признака в таблице. — В сб. «Вопросы кибернетики». Ташкент, Изд-во Ин-та кибернетики АН Узб. ССР, 1970. № 37.