

На правах рукописи

Романов Михаил Юрьевич

**Построение обобщённых полиномов
минимальной степени
над алгоритмами вычисления оценок**

Специальность 05.13.17 — теоретические основы информатики

Автореферат
диссертации на соискание учёной степени
кандидата физико-математических наук

Москва
2008

Работа выполнена в Вычислительном центре им. А. А. Дородницына Российской Академии наук.

Научный руководитель: доктор физико-математических наук,
профессор, академик РАН
Журавлёв Юрий Иванович.

Официальные оппоненты: доктор физико-математических наук,
профессор, чл.-корр. РАН
Рудаков Константин Владимирович,
кандидат физико-математических наук
Кольцов Пётр Петрович.

Ведущая организация: Московский государственный университет
им. М. В. Ломоносова.

Защита диссертации состоится «___» _____ 2008 г. в ___ часов на заседании диссертационного совета Д002.017.02 в Вычислительном центре им. А. А. Дородницына Российской Академии Наук по адресу: 119333, Москва, ул. Вавилова, 40.

С диссертацией можно ознакомиться в библиотеке ВЦ РАН.

Автореферат разослан «___» _____ 2008 г.

Учёный секретарь
диссертационного совета
Д002.017.02 при Вычислительном центре
им. А. А. Дородницына РАН
доктор физико-математических наук, профессор

Рязанов В.В.

Общая характеристика работы

Актуальность темы. Теория распознавания является одной из очень важных и динамично развивающихся областей прикладной математики. Её методы позволяют формулировать и успешно решать задачи из таких прикладных областей, как экономика, социология, геология, химия, биология и медицина.

К концу 50-х годов прошлого века были сформулированы многие параметрические семейства алгоритмов распознавания. Ю. И. Журавлёвым разработан класс алгоритмов вычисления оценок (АВО), который также можно использовать в качестве языка описания методов распознавания. Следующим важным этапом развития теории распознавания стало создание Ю. И. Журавлёвым алгебраического подхода, который является эффективным способом описания алгоритмических композиций. В рамках этого подхода наибольший интерес представляют полиномиальные композиции. В частности, было показано, что для любой регулярной задачи существует корректный алгоритм среди полиномов над АВО. Это привело к широкому применению полиномиальных алгоритмических структур при исследовании и решении прикладных проблем. Поэтому одной из ключевых задач стало уменьшение сложности алгебраических композиций. Значительное количество работ посвящено оценке степени и числа слагаемых корректного полинома над алгоритмами распознавания. Этой задачей занимались Ю. И. Журавлёв, И. В. Исаев, В. Л. Матросов, Т. В. Плохонина, К. В. Рудаков, А. С. Дьяконов. В настоящей диссертации продолжены эти исследования и приведён метод построения полинома минимальной степени над заданным набором алгоритмов.

Цель работы состоит в разработке метода построения обобщённого полиномиального алгоритма наименьшей степени в алгебре над множеством алгоритмов вычисления оценок.

Методика исследований. В диссертации использован аппарат алгебраического подхода к теории распознавания. Также использованы методы теории оптимизации.

Научная новизна. Все результаты, полученные в диссертации, являются новыми. Впервые рассмотрено понятие обобщённого полиномиального алгоритма, в котором степени операторов рассматриваются из множества действительных чисел. Предложен метод решения оптимизационной задачи, определяющей степени операторов в обобщённом полиноме, используя декомпозицию задачи на множество более простых задач. В диссертации показан метод решения вспомогательных оптимизационных задач, являющийся альтернативным методу последовательного квадратичного программирования, и являющийся более эффективным.

В работе предложены подходы, повышающие эффективность предложенного метода. Показано решение задачи уменьшения числа слагаемых алгебраического полинома для понижения его степени. Также в диссертации рассмотрен вопрос многопроцессорной реализации приведённых методов.

Апробация работы. Результаты, изложенные в диссертации, докладывались на Всероссийской конференции «Математические методы распознавания образов» XIII (Москва, 2007 г.); «Интеллектуализация обработки информации»–2008 (Алушта, Украина, 2008 г.); а также на научных семинарах МФТИ.

Основные результаты диссертации.

- 1) Введено понятие обобщённого полиномиального алгоритма в алгебре над множеством алгоритмов вычисления оценок.
- 2) Для нахождения обобщённого полиномиального алгоритма минимальной степени сформулирована оптимизационная задача.
- 3) Предложен метод декомпозиции рассмотренной оптимизационной задачи на множество вспомогательных задач.
- 4) Построен алгоритм решения вспомогательных задач.

- 5) Рассмотрены два подхода повышения эффективности предложенного метода. Первый подход позволяет сократить число решаемых вспомогательных задач, второй — снизить их сложность.
- 6) Приведено развитие этого подхода, позволяющее понизить степень обобщённого полинома за счёт уменьшения числа слагаемых (распознающих операторов).
- 7) Рассмотрен вопрос многопроцессорной реализации приведённых методов.
- 8) Проведена серия экспериментов по практической реализации предложенного метода. На рассмотренных примерах показана высокая эффективность метода.
- 9) Проведены практические эксперименты для сравнения эффективности со стандартными методами решения рассмотренной оптимизационной задачи. Показано существенное превосходство предложенного метода.

Практическая и теоретическая ценность. Работа носит в основном теоретический характер. Введённое понятие обобщённого полиномиального алгоритма может использоваться, как естественное расширение классических алгебраических полиномиальных алгоритмов. Методы, представленные в диссертации, могут быть непосредственно использованы на практике, а также служить основой для дальнейших теоретических исследований полиномиальных корректирующих операций. Эффективность предложенных подходов подтверждена решением практических задач.

Публикации. По теме диссертации автором опубликованы 4 работы.

Структура и объём работы. Работа состоит из введения, четырёх глав и списка литературы из 33 наименований. Общий объём работы — 80 страниц, включая 4 рисунка и 27 таблиц.

Содержание диссертационной работы

Во **введении** даётся краткий обзор полученных ранее результатов. Приводится общая структура диссертации и краткое изложение содержания работы.

В **первой главе** вводятся основные понятия и определения используемые в работах Ю. И. Журавлёва, посвященных алгебраическому подходу к решению задач распознавания и классификации. Рассматриваются алгоритмы вычисления оценок (АВО). Описывается формальная постановка задачи построения распознающего алгоритма следующим образом.

Множество из M объектов разделяется на *обучающую выборку* \tilde{S}_m и *контрольную выборку* \tilde{S}^q .

Задача распознавания $Z = (I_0, \tilde{S}^q)$ определяется начальной информацией I_0 о принадлежности объектов классам и контрольной выборкой \tilde{S}^q . Для каждого контрольного вектора необходимо найти информационный вектор

$$\tilde{\beta}(S^i) = \left(\beta_{ij} = P_j(S^i) \equiv (S^i \in K_j), j = \overline{1, l} \right).$$

Рассматривается множество алгоритмов $\{A\}$ решения задачи $Z = (I_0, \tilde{S}^q)$, представленных в виде произведения операторов $A = B \cdot C$. Здесь оператор B определяет матрицу оценок $B(Z) = \|\Gamma_{ij}\|_{q \times l}$, где Γ_{ij} — действительные числа; а оператор C — матрицу ответов

$$C\left(\|\Gamma_{ij}\|_{q \times l}\right) = \|\beta_{ij}\|_{q \times l}, \text{ где } \beta_{ij} \in \{0, 1, \Delta\}.$$

Возможность представления произвольного алгоритма распознавания в таком виде показана Ю. И. Журавлёвым.

Определение 1.1. Алгоритм A называется **корректным** для задачи Z , если выполнено равенство $A(I_0, \tilde{S}^q) = \|\alpha_{ij}\|_{q \times l}$. Алгоритм A , не являющийся корректным для Z , называется **некорректным**.

Определение 1.2. Решающее правило C называется **корректным** на M , если для всякого конечного набора $\tilde{S}_m \in M$ существует хотя бы одна числовая матрица $\|a_{ij}\|_{q \times l}$ такая, что $C\left(\|a_{ij}\|_{q \times l}\right) = \|\alpha_{ij}\|_{q \times l}$.

Для распознающих операторов (РО) можно ввести операции сложения, умножения на константу и операторного умножения таким образом, что это будут операции над оценками, то есть над значениями операторов. Эти операции индуцируют соответствующие операции над алгоритмами распознавания. Это позволяет ввести над множеством РО операции линейного и алгебраического замыкания.

Определение 1.3. Множества $L(\{A\})$, $U(\{A\})$ алгоритмов $A = B \cdot C$ соответственно таких, что $B \in L(\{B\})$, $B \in U(\{B\})$, называется соответственно **линейным** и **алгебраическим замыканиями** $\{A\}$.

Множество алгоритмов, построенное на алгебраическом замыкании k -й степени над некоторым набором РО, будем называть алгебраическим расширением k -й степени.

Для краткости степенью алгоритма A будем называть степень алгебраического расширения, в котором построен этот алгоритм. Иначе говоря, это можно понимать, как степень полинома над операторами $\{B_k\}$, который является распознающим оператором алгоритма A .

В работах Ю. И. Журавлёва получены следующие результаты.

Теорема 1.1. Пусть $\{A\}$ — совокупность некорректных алгоритмов, $\{B\}$ — соответствующее множество операторов, C — фиксированное корректное решающее правило. Тогда $L\{A\} = \{L\{B\} \cdot C\}$ является корректным относительно $\{Z\}$, если $\{Z\}$ состоит из задач, полных относительно $\{B\}$.

Теорема 1.2. Пусть выполнены все условия предыдущей теоремы и, кроме того, $[\{B\}]$ есть замыкание $\{B\}$ относительно операций умножения на константу и операторного умножения. Тогда $U\{A\} = \{U\{B\} \cdot C\}$ является корректным относительно $\{Z\}$, если $\{Z\}$ состоит из задач, полных относительно $[\{B\}]$.

Для алгоритмов вычисления оценок (АВО) предполагается, что элемент матрицы распознающего оператора B равен $\Gamma_{ij} = \Gamma_j(S^i)$ и является оценкой объекта S^i по классу K_j .

В качестве решающего правила используется стандартное корректное пороговое решающее правило: $C(\|\Gamma_{ij}\|_{q \times l}) = \|C(\Gamma_{ij})\|_{q \times l}$,

$$C(\Gamma_{ij}) = \left\{ 0, \Gamma_{ij} \leq c_1; 1, \Gamma_{ij} \geq c_2; \Delta, c_1 < \Gamma_{ij} < c_2 \right\}.$$

Элементы матрицы ответов можно разбить на два множества $\|\beta_{ij}\|_{q \times l} = M_0 \cup M_1$, где $M_i = \{(u, v) : \beta_{uv} = i\}$, q — число объектов, l — число классов.

Оператор B с матрицей оценок $B(Z) = \|\Gamma_{ij}\|_{q \times l}$ называется *допустимым* для задачи Z , если существует хотя бы одна пара $(u, v) \in M_1$ такая, что для всех $(i, j) \in M_0$ выполняется $\Gamma_{uv}(B) > |\Gamma_{ij}(B)|$.

Такая пара (u, v) называется *отмеченной* в B , и множество отмеченных в B пар обозначим через $M(B)$. Фактически, это означает, что отмеченные пары «отделены» от всех пар (i, j) таких, что объект S^i не принадлежит классу K_j . Поэтому, при некоторых порогах c_1, c_2 алгоритм $B \cdot C$ будет давать правильные ответы для всех пар, отмеченных в B .

Допускается альтернативный способ определения отмеченных точек. Пара (u, v) называется *отмеченной* в B при некоторой фиксированной величине δ , если для всех $(i, j) \in M_0$ имеет место неравенство

$$\Gamma_{uv}(B) \geq |\Gamma_{ij}(B)| + \delta.$$

При таком определении все дальнейшие рассуждения остаются справедливыми.

$$\text{Пусть } \Gamma_{\max}^0(B) = \max_{(i,j) \in M_0} |\Gamma_{ij}(B)|, \quad \Gamma_{\min}^1(B) = \min_{(i,j) \in M(B)} |\Gamma_{ij}(B)|.$$

$$\text{Введём } \Gamma(B) = \left[\Gamma_{\min}^1(B) \right]^{-1}, \quad Q(B) = \Gamma_{\max}^0(B) / \Gamma_{\min}^1(B).$$

Для оператора B можно ввести оператор $B' = \Gamma(B) \cdot B$. Определив в задаче $Z = (I_0, \tilde{S}^q)$ элементы матрицы оценок, как $B'(Z) = \|\Gamma'_{ij}\|_{q \times l}$, получим $\Gamma'_{ij}(B) = \Gamma(B) \cdot \Gamma_{ij}(B)$.

Ю. И. Журавлёвым показано следующее утверждение.

Лемма 1.3. *Если (u, v) отмечена в B , то $\Gamma'_{ij}(B) \geq 1$; если (u, v) не отмечена в B , то $\Gamma'_{ij}(B) \leq Q(B) < 1$.*

Тогда, учитывая, что матрица оценок $B'(Z)$ получается из $B(Z)$ поэлементным умножением, получим, что пара (u, v) является отмеченной в B' тогда и только тогда, когда она отмечена в B .

Пусть $\{B_k\}$ — произвольная конечная система распознающих операторов.

Определение 1.4. Система $\{B_k\}$ является **базисной** для Z , если $M_1 = \bigcup_{k=1}^n M(B_k)$.

В работе Ю. И. Журавлёва приводится теорема о том, что полиномиальный алгоритм со степенями, определяемые некоторой формулой, является корректным для Z . При этом, указанные в теореме степени распознающих операторов оказываются заведомо большими, чем необходимо для выполнения условий корректности.

В работах К. В. Рудакова и А. Г. Дьяконова получена точная верхняя грань степени полинома, необходимой для корректности алгоритма. Для любого набора операторов, если существует корректный полиномиальный алгоритм, мы можем полагать, что степень этого алгоритма не превосходит величину $\lceil \log_2 ql \rceil$.

В **главе 2** формулируется основная оптимизационная задача и описывается метод её решения.

В параграфе 2.1 рассматривается обобщение введённых полиномиальных алгоритмов. В отличие от предыдущего, степень распознающего оператора предполагается действительным числом, а не только целым.

Определение 2.1. **Обобщённым алгебраическим замыканием** множества распознающих операторов $B^* = \{B_1, B_2, \dots, B_n\}$ будем называть множество

$$\bar{U}(B^*) = L\left(\{B_1^{x_1} \cdot B_2^{x_2} \cdot \dots \cdot B_s^{x_s} : B_i \in B^*, x_i \in \mathbb{R}, i = \overline{1, s}, 1 \leq s \leq k\}\right).$$

Элементы этого множества назовём **обобщёнными полиномами** над B^* , а слагаемые этого полинома — **обобщёнными мономами**.

Аналогично, множество $\bar{U}(A^*)$ алгоритмов $A = B \cdot C$ таких, что $B \in \bar{U}(B^*)$, назовём **обобщённым алгебраическим замыканием** A^* .

Определение 2.2. Пусть задано множество PO

$$B^* = \{B_1, B_2, \dots, B_n\}.$$

Степенью обобщённого монома $B_1^{x_1} \cdot B_2^{x_2} \cdot \dots \cdot B_s^{x_s}$ будем называть величину $x_1 + x_2 + \dots + x_s$.

Степенью обобщённого полинома $V \in \bar{U}(B^*)$ будем называть максимальную из степеней входящих в него обобщённых мономов.

Степенью алгоритма $A = B \cdot C$ из обобщённого алгебраического замыкания будем называть степень обобщённого полинома B .

Всякий полином над распознающими операторами является также и обобщённым полиномом. При этом степень полинома совпадает со степенью, полученной при рассмотрении этого полинома, как обобщённого.

Для введённых понятий сохраняются результаты главы 1. В частности, приведённая в параграфе 1.3 точная верхняя грань степени полинома означает, что существует обобщённый полином указанной степени.

В параграфе 2.2 рассматривается построение распознающего алгоритма в обобщённом алгебраическом замыкании над множеством алгоритмов вычисления оценок. Для нахождения искомого набора степеней сформулируем некоторую оптимизационную задачу.

Пусть задан набор РО B_k , строящих матрицы оценок $B_k(Z) = \|\Gamma_k^{uv}\|_{q \times l}$, и оператор C задан стандартным пороговым решающим правилом: $C(\Gamma_k^{uv}) = \left\{ 0, \Gamma_k^{uv} \leq c_1; 1, \Gamma_k^{uv} \geq c_2; \Delta, c_1 < \Gamma_k^{uv} < c_2 \right\}$, которое определяется параметрами c_1, c_2 .

Замечание 2.1. Без ограничения общности будем считать, что для параметров c_1, c_2 выполнено условие $c_1 + c_2 = 1$. В противном случае, эти параметры следует разделить на сумму $c_1 + c_2$.

Определение 2.3. Набор распознающих операторов $\{B_k\}$ будем называть **правильным**, если для него выполняются следующие условия.

- 1) Каждый оператор является допустимым, т. е. существует хотя бы одна пара $(u, v) \in M_1$ отмеченная в B .
- 2) Каждый оператор является нормированным, т. е. по лемме 1.3 если (u, v) отмечена в B , то $\Gamma_k^{uv} \geq 1$, иначе $\Gamma_k^{uv} < 1$.
- 3) Система $\{B_k\}$ является базисной для Z , т. е. $M_1 = \bigcup_{k=1}^n M(B_k)$.

В противном случае набор операторов будет называться **неправильным**.

В дальнейшем описывается алгоритм, корректный для задачи Z , в следующем виде: $A = \left\{ \sum_{k=1}^n B_k^{x_k} \right\} \circ C(c_1, c_2)$. Сформулируем теорему, которая накладывает условия на степени обобщённых полиномов, при которых рассматриваемый алгоритм является корректным.

Теорема 2.1. Пусть задан набор операторов $\{B_k\}$, $k = \overline{1, n}$, дающих матрицы оценок $B_k(Z) = \|\Gamma_k^{uv}\|_{q \times l}$ на обучающей выборке задачи Z .

Пусть для набора степеней $\{x_k\}$ имеют место следующие условия:

- 1) для любой пары $(u, v) \in M_0$ выполняется $\Gamma_{uv} = \sum_{k=1}^n (\Gamma_k^{uv})^{x_k} \leq c_1$;
- 2) для любой пары $(u, v) \in M_1$ выполняется $\Gamma_{uv} = \sum_{k=1}^n (\Gamma_k^{uv})^{x_k} \geq c_2$.

Тогда алгоритм

$$A = \left\{ \sum_{k=1}^n B_k^{x_k} \right\} \circ C(c_1, c_2) \quad (1)$$

является корректным для Z .

Пусть также набор операторов $\{B_k\}$ является правильным по определению 2.3, т.е. операторы являются допустимыми, нормированными и образуют базисную систему для Z . Тогда существует набор степеней $\{x_k\}$ определяющий алгоритм A , корректный для задачи Z .

Теперь сформулируем оптимизационную задачу для определения набора степеней $\{x_k\}$ удовлетворяющих условиям теоремы 2.1 и дающих обобщённый полином минимальной степени:

$$\begin{cases} \forall (u, v) \in M_0: & \sum_{k=1}^n (\Gamma_k^{uv})^{x_k} \leq c_1; \\ \forall (u, v) \in M_1: & \sum_{k=1}^n (\Gamma_k^{uv})^{x_k} \geq c_2; \\ \max_{k=\overline{1, n}} x_k \rightarrow \min. \end{cases} \quad (2)$$

В полученной оптимизационной задаче (2) область ограничений состоит из $q \times l$ неравенств, и целевая функция зависит от n переменных.

Введём замену переменных: $y_k = e^{x_k}$, $\gamma_k^{uv} = \ln \Gamma_k^{uv}$, $\varphi_{uv}(\tilde{y}) = \sum_{k=1}^n y_k \gamma_k^{uv}$. Тогда, учитывая результаты параграфа 1.3, оптимизационная задача (2) принимает вид

$$U = \left\{ \tilde{y} \left| \begin{array}{l} \varphi_{uv}(\tilde{y}) \leq c_1, \text{ для всех } (u, v) \in M_0; \\ \varphi_{uv}(\tilde{y}) \geq c_2, \text{ для всех } (u, v) \in M_1; \\ 0 \leq y_k \leq M = ql, \text{ для всех } k = \overline{1, n} \end{array} \right. \right\}. \quad (3)$$

$$R = \left\{ \tilde{y} \in U, \max_{k=\overline{1, n}} y_k \rightarrow \min \right\}. \quad (4)$$

В параграфе 2.3 описывается метод декомпозиции рассматриваемой оптимизационной задачи. Для каждого непустого множества индексов $I = \{i_1, \dots, i_m\} \subseteq \{1, \dots, n\}$ рассматривается задача

$$R_I = \{ \tilde{y} \in U_I, y_{i_1} \rightarrow \min \} \quad (5)$$

с множеством ограничений

$$U_I = \{ \tilde{y} \in U; y_{i_1} = y_{i_2} = \dots = y_{i_m}; y_{i_1} \geq y_k, k \notin I \}. \quad (6)$$

Таким образом, исходной задаче R ставится в соответствие $2^n - 1$ вспомогательных задач R_I . В последующих параграфах будет показано, что задачу R с нелинейным функционалом можно решить, найдя решение более простых задач R_I с линейным функционалом. В следующей главе будут приведены методы, позволяющие существенно сократить количество решаемых задач.

В параграфе 2.4 рассматривается геометрическая интерпретация предложенного метода декомпозиции.

В параграфе 2.5 приводится теорема о существовании решения оптимизационной задачи.

Теорема 2.2. *Существует непустое множество индексов*

$$I = \{i_1, \dots, i_m\} \subseteq \{1, \dots, n\}$$

такое, что решение задачи R_I является решением задачи R .

В параграфе 2.6 рассмотрены алгоритмы решения вспомогательных задач R_I . Описан метод линеаризации, позволяющий свести задачу к последовательности задач квадратичного программирования. Так как все ограничения множества U_I являются полиномиальными, то такое сведение удаётся сделать аналитически.

Далее описаны алгоритмы решения задачи квадратичного программирования методом линеаризации и обобщённым методом Ньютона.

В настоящий момент наиболее эффективным методом решения задачи нелинейной оптимизации (задача R_I) считается метод последовательного квадратичного программирования (Sequential Quadratic Programming, или SQP). Он позволяет сводить исходную нелинейную задачу к последовательности задач квадратичного программирования, на каждом шаге осуществляя аппроксимацию Гессiana функции Лагранжа с помощью обобщённого метода Ньютона.

В предложенном подходе также используется сведение нелинейной задачи R_I к последовательности задач квадратичного программирования. В отличие от метода SQP для сведения применяется не квадратичная аппроксимация функции Лагранжа, а метод линеаризации. Несмотря на высокую эффективность метода SQP на многих задачах, для рассматриваемой задачи метод линеаризации оказывается существенно эффективнее.

Подробнее вопрос сравнения эффективности на прикладных задачах рассматривается в параграфе 4.4.

В **главе 3** рассматриваются способы, позволяющие существенно повысить эффективность метода, описанного в главе 2. Приводится развитие этого подхода, позволяющее понизить степень обобщённого полинома, уменьшая число слагаемых.

Для описания свойств вспомогательных оптимизационных задач вводятся некоторые специальные операторы. Пусть для некоторой оптимизационной задачи R_I при наборе $I = \{i_1, \dots, i_m\} \subseteq \{1, \dots, n\}$, решением является вектор $\tilde{y}(R_I)$ размерности n . Тогда для задачи R_I введём следующие обозначения:

$$F_I = \max_{k=\overline{1,n}} \tilde{y}(R_I) \Big|_k; \quad G_I = \tilde{y}(R_I) \Big|_{i_1}. \quad (7)$$

Для задачи R с решением \tilde{y}^* введём $F(R) = F^* = \max_{k=\overline{1,n}} y_k^*$. При этом для любого набора I выполняется $G_I \leq F_I$. Сформулируем несколько свойств для введённых операторов.

Теорема 3.1. Пусть заданы два набора индексов $I^1 \subset I^2$, где

$$I^1 = \{i_1^1, \dots, i_{m_1}^1\} \subseteq \{1, \dots, n\},$$

$$I^2 = \{i_1^2, \dots, i_{m_2}^2\} \subseteq \{1, \dots, n\}.$$

Тогда: 1) $U_{I^1} \supset U_{I^2}$; 2) $G_{I^1} \leq G_{I^2}$.

Теорема 3.2. Пусть для набора $I^* = \{i_1^*, i_2^*, \dots, i_{m_1}^*\} \subseteq \{1, 2, \dots, n\}$, решение задачи R_{I^*} совпадает с решением задачи R . Тогда для любого набора $I = \{i_1, i_2, \dots, i_{m_1}\} \subseteq \{1, 2, \dots, n\}$, выполняется $F_{I^*} \leq F_I$.

Замечание 3.1. В теореме 2.2 решение задачи R сводится к выбору среди решений задач R_I . Таким образом, необходимо найти набор I^* , определяющий вспомогательную задачу, решение которой является решением задачи R . Тогда, по теореме 3.2 достаточно искать минимум величины $\max_{k=\overline{1,n}} y_k$ не на всём множестве U , а лишь среди всех решений задач R_I . Для них эта величина равна F_I , и в новых обозначениях задачу минимизации можно сформулировать следующим образом:

$$I^* = \arg \min_{I \subseteq \{1, \dots, n\}} F_I.$$

Лемма 3.3. Пусть для некоторых I^1 и I^2 выполняется следующее свойство: $G_{I^1} \geq F_{I^2}$. Тогда для любого набора I , содержащего I^1 , решение задачи R_I является решением задачи R только тогда, когда задача R_{I^2} даёт решение задачи R .

Замечание 3.2. Из доказанной леммы 3.3 следует, что если найдены наборы I^1 и I^2 , удовлетворяющие условию леммы, то все I , содержащие поднабор I^1 , следует исключить из рассмотрения. Это утверждение лежит в основе рассматриваемого далее алгоритма эффективного перебора вспомогательных задач.

Описывается эффективный **алгоритм поиска** вспомогательной задачи R_{I^*} , дающей решение задачи R .

АЛГОРИТМ.

- 1) Рассматриваем упорядоченное множество наборов

$$H = \{I: I \subseteq \{1, 2, \dots, n\}; I \neq \emptyset\},$$

которое упорядочено таким образом, что наборы с меньшей мощностью предшествуют наборам с большей. Присваиваем начальное значение $F_{\min} = +\infty$.

- 2) Выбираем первый нерассмотренный набор I из H и решаем задачу R_I .
- 3) Вычисляем значения G_I и F_I . Если $F_I < F_{\min}$, то присваиваем $F_{\min} = F_I$.
- 4) Если $G_I \geq F_{\min}$, то удаляем из H все наборы, которые содержат поднабор I (см. замечание 3.2).
- 5) Если в H рассмотрены не все наборы, то переходим к шагу 2.
- 6) Искомое решение, согласно теореме 3.2 и замечанию 3.1, определяется тем набором I^* , для которого $F_{I^*} = F_{\min}$.

В параграфе 3.2 приводится следующий метод сокращения общего количества вычислений. Множество U (см. формулу (3)) содержит ограничения:

$$0 \leq y_i \leq M, \quad i = \overline{1, n},$$

где полагается $M = ql$, согласно результатам параграфа 1.3.

В алгоритме, описанном в параграфе 3.1 на шаге 3 производится корректировка значения F_{\min} . При этом можно после каждого шага брать $M = F_{\min}$. Тогда на каждом шаге область ограничений не увеличивается, а в некоторых случаях уменьшается. Кроме того, при фиксированном уровне точности такой подход может позволить заметно сократить количество итераций за счет того, что, начиная с некоторого шага, линейные размеры области ограничений окажутся меньше величины ошибки.

Замечание 3.3. Во многих случаях оказывается полезным в первую очередь решить задачу R_I при $I = \{1, \dots, n\}$, тем самым уже на первом шаге существенно уменьшив область ограничений.

В экспериментах, приведённых в параграфе 4.3, наиболее существенное уменьшение области ограничений происходило именно на таком первом шаге.

В параграфе 3.3 описана модификация приведённого алгоритма для эффективной работы на многопроцессорных системах.

В параграфе 3.4 рассматривается возможность уменьшения степени полиномиального алгоритма за счёт уменьшения количества РО.

Рассмотрим начальный набор распознающих операторов $\mathfrak{B} = \{B_k\}$, который является базисным для задачи распознавания Z . Решение задачи $R(\mathfrak{B})$ дает алгоритм $A(\mathfrak{B})$, корректный для Z , причем степень обобщённого полинома запишем, как $F(\mathfrak{B}) = F(R(\mathfrak{B}))$.

Определение 3.1. Пусть для набора \mathfrak{B}' заданы функции $\varphi'_{uv}(\tilde{y})$, определяющие ограничения оптимизационной задачи. Будем говорить, что:

- набор $\mathfrak{B}' \subseteq \mathfrak{B}$ имеет тип 1, если для некоторой пары $(u, v) \in M_1$ выполняется $\varphi'_{uv}(\tilde{y}) = c_2$;
- набор $\mathfrak{B}' \subseteq \mathfrak{B}$ имеет тип 0, если для некоторой пары $(u, v) \in M_0$ выполняется $\varphi'_{uv}(\tilde{y}) = c_1$.

Теорема 3.4. Из распознающих операторов набора \mathfrak{B} обобщённый полином наименьшей степени может быть образован либо всем набором \mathfrak{B} , либо поднабором $\mathfrak{B}' \subset \mathfrak{B}$, соответствующим тупиковому покрытию множества M_1 . При этом \mathfrak{B} является оптимальным тогда и только тогда, когда имеет тип 1.

Из теоремы 3.4 следует, что достаточно решить задачу R для набора \mathfrak{B} и всех его тупиковых (или базисных) поднаборов.

В параграфе 3.5 приводится адаптация методов увеличения эффективности для случая минимизации числа слагаемых.

Описывается алгоритм нахождения подмножества исходного набора распознающих операторов \mathfrak{B} , дающего корректный обобщённый полином минимальной степени.

АЛГОРИТМ.

- 1) Решаем задачу $R(\mathfrak{B})$. По определению 2.3 проверяем тип набора \mathfrak{B} . Если получен тип 1, то задача решена. Иначе, переходим на следующий шаг.
 - 2) Для каждого оператора из набора $\mathfrak{B} = \{B_k\}$ находим множество $M(B_k)$ — множество отмеченных пар.
 - 3) Находим все тупиковые покрытия множества M_1 множествами $M(B_k)$, $k = \overline{1, n}$. Пронумеруем наборы, соответствующие покрытиям $\{\mathfrak{B}^i, i = \overline{1, N}\}$.
 - 4) Последовательно решаем задачи $R(\mathfrak{B}^i)$, и среди полученных решений выбираем то, которое даёт наименьшую степень.
-

Аналогично результатам параграфа 3.2 описан метод упрощения перебора тупиковых, основанный на уменьшении ограничения M . Пусть на p -й итерации минимальным решением является $F_{\min}^p = F(\mathfrak{B}^s)$, $s \leq p$, тогда при решении задачи $R(\mathfrak{B}^{p+1})$ мы можем положить $M = F_{\min}^p$. Таким образом, мы на каждом шаге будем сужать область ограничения последующих задач, за исключением тех шагов, где область ограничения окажется пуста. Здесь также возможно применение приближенного подхода.

Описан подход к эффективному распараллеливанию рассматриваемого алгоритма, аналогичный приведённому в параграфе 3.3.

В **главе 4** приведены результаты экспериментального исследования рассмотренных методов.

Для практических задач ограничения на набор распознающих операторов, наложенные в определении 2.3, являются очень жёсткими. Поэтому в параграфе 4.2 описывается расширение предложенного метода, позволяющее строить обобщённые полиномиальные корректоры над более широким классом распознающих операторов.

В параграфе 4.3 приведены практические эксперименты. В качестве источника данных используются задачи из репозитория UCI.

Для практического исследования работы алгоритма нахождения обобщённого полинома минимальной степени на заданном наборе из n распознающих операторов, этот алгоритм реализован на языке MATLAB. Каждая вспомогательная задача сводится методом линеаризации к последовательности задач квадратичного программирования (см. параграф 2.6.1). В качестве процедуры решения задачи квадратичного программирования используется метод `quadprog` из пакета `Optimization Toolbox`, который основан на обобщённом методе Ньютона (см. параграф 2.6.3). Для обработки случая, когда набор операторов не является базисом, используются результаты параграфа 4.2. В качестве первой вспомогательной задачи рассматривается задача R_I при $I = \{1, \dots, n\}$ для уменьшения величины M (см. замечание 3.3). Затем применяется алгоритм сокращения перебора вспомогательных задач (см. параграф 3.1).

При решении рассматриваемых задач используется набор алгоритмов из пакетов `MatLabArsenal` и `WEKA`.

Предложенный алгоритм сравнивается в режиме скользящего контроля с алгоритмом голосования.

РЕЗУЛЬТАТЫ СКОльзяЩЕГО КОНТРОЛЯ (в %)				
Задача	Iris	Sonar	Musk	Breast Cancer
$A = \left\{ \sum_{k=1}^n B_k^{x_k} \right\} \cdot C$	2.0	9.6	17.84	3.51
Голосование	7.3	14.9	18.46	3.87
УМЕНЬШЕНИЕ ОБЛАСТИ ОГРАНИЧЕНИЙ				
$M = ql$	450	416	952	1138
M после первого шага	0.59	0.58	0.59	0.38
$\max_{k=1,n} y_k$	0.53	0.53	0.53	0.34

Для всех рассмотренных задач можно отметить преимущество обобщённого полиномиального алгоритма над алгоритмом голосования. Во всех экспериментах наиболее существенное уменьшение области ограничений происходило на первом шаге. Кроме того, метод сокращения числа решаемых оптимизационных задач позволил уменьшить их количество с $2^n - 1$ до $n + 1$.

В параграфе 4.4 проведено сравнение эффективности методов решения задачи, описанных в параграфе 2.6.

В результате применение декомпозиции исходной задачи существенно уменьшает время её решения. Фактически, после этой операции время решения становится пригодным для практического использования. А применение метода линеаризации для перехода к задаче квадратичного программирования и использование подходов главы 3 позволяет ещё раз заметно повысить эффективность алгоритма.

Список публикаций по теме диссертации

1. *Романов М. Ю.* Об одном методе построения распознающего алгоритма в алгебре над множеством вычисления оценок // Ж. вычисл. матем. и матем. физ. — 2007. — Т. 47, № 8. — С. 1426–1430.
2. *Романов М. Ю.* Построение корректного распознающего алгоритма минимальной степени в алгебре над множеством алгоритмов вычисления оценок // ММРО-13. — М.: МАКС Пресс, 2007. — С. 60–62.
3. *Романов М. Ю.* Реализация одного метода построения распознающего алгоритма в алгебре над множеством алгоритмов вычисления оценок // Ж. вычисл. матем. и матем. физ. — 2008. — Т. 48, № 9.
4. *Романов М. Ю.* Полиномиальный корректный распознающий алгоритм минимальной степени в алгебре над множеством алгоритмов вычисления оценок // ИОИ-2008. — Симферополь, 2008.