УДК 519.8 doi: 10.15421/322009

И.В. Козин, Е.К. Селютин
Запорожский национальный университет

МЕТАЭВРИСТИКИ ДЛЯ ПОИСКА ОПТИМАЛЬНЫХ КЛАССИФИКАЦИЙ

Исследуется проблема поиска оптимальных классификаций на конечном множестве. Показано, что задача поиска оптимальной классификации, порождаемой отношением толерантности на конечном множестве, сводится к задаче оптимизации на множестве перестановок. Предложены метаэвристики для поиска субоптимальных решений задачи классификации.

Ключевые слова: задача оптимальной классификации, классификация, эволюционный алгоритм, алгоритм муравьиной колонии, алгоритм перемешанных прыгающих лягушек.

І.В. Козін, Є.К. Селютін
Запорізький національний університет

МЕТАЕВРИСТИЧНІ МЕТОДИ ПОШУКУ ОПТИМАЛЬНИХ КЛАСИФІКАЦІЙ

Одним з головних завдань будь-якого наукового дослідження є задача класифікації. Хоча формалізація задачі класифікації є досить простою, рішення її може виявитися досить складним в обчислювальному сенсі. Більшість існуючих методів класифікації засновано на статистичних принципах, факторизації, кластерному та дисперсійному аналізі. Часто для побудови «хороших» класифікацій використовують системи, що здатні навчатись, зокрема, нейронні мережі. Складність завдання класифікації зумовлена невизначеністю розбиття на класи і складністю пошуку оптимальної класифікації за умов використання певного критерію оптимальності.

Зазначено, що задача класифікації на кінцевій множині полягає в пошуку розбиття, що володіє деякими заданими властивостями. Розбиття множини задає канонічне відношення еквівалентності, пов'язане з цим розбиттям. Зокрема, два елементи вважаються еквівалентними, якщо вони належать одному елементу розбиття. З іншого боку, будь-яке відношення еквівалентності на кінцевій множині визначає його розбиття на класи еквівалентних між собою елементів

У цій статті розглядаються класифікації, які побудовано на основі відносини толерантності або близькості. При цьому розбиття на класи є неоднозначним і істотно залежить від вибору представників класів.

В роботі показано, що задача пошуку оптимальної класифікації на скінченій множині за умов наявності критерію оптимальності може бути зведена до задачі пошуку оптимальної перестановки елементів цієї множини.

Задача пошуку оптимальної перестановки відноситься до розряду важких в обчислювальному сенсі задач. У статті пропонується метод використання деяких відомих метаевристик до задачі пошуку субоптимальних перестановок елементів множини за заданим критерієм оптимальності. Розглянуто три метаевристики на перестановках елементів скінченої множини (еволюційний алгоритм, алгоритм мурашиної колонії і

© Козин И.В., Селютин Е.К., 2020

метод перемішаних стрибаючих жаб), які можна з успіхом застосовувати при вирішенні задачі пошуку оптимальної класифікації.

Ключові слова: задача оптимальної класифікації, класифікація, еволюційний алгоритм, алгоритм мурашиної колонії, алгоритм перемішаних стрибаючих жаб.

I.V. Kozin, E.K. Seliutin
Zaporizhia National University

THE METAHEURISTICS FOR OPTIMAL CLASSIFICATIONS SEARCHING

One of the most important science problems is the task of classification. While formalizing the tasks of classifying can be completed with downtime, its solution can be completed by decimal sense. Most of the classification methods are based on statistical principles, factorization, cluster analysis and dispersion analysis. Often, to incite the "good" classic victorious systems, what's the most healthiest, secrecy, neural measure. Classification task foldability is enriched by the unapproachability of class.

It has been designated that classification task on the finite set is the task of dividing. Dividing of the multitude of tasks is the canonical presentation of equivalence, due to it. For example, two elements get involved in equivalents, as if they stink with one divided element. From the first side, be the first and the same equivalency to the largest number of cards in the class of equal elements.

At these paper the basis of the principle of tolerance and closeness of classification were analyzed.

It is shown in the robot that the task of making optimal classifications for the final number of brains is obvious criteria for optimality, but it is possible to set the task of making optimal permutation of elements of multiple values.

The task of making an optimal rearrangement is to be brought up to the list of important ones in the calculus of sensation. The statistics have the method of recognizing the viable ideas of meta-tasks for the task of making suboptimal permutations of elements of a set for a given criterion of optimality. Three meta-statistics on permutations of elements of the largest number (European algorithm, algorithm of colony and method of mixing shaving toads), which can be accomplished with success in solving tasks in an optimal way, are considered.

Keywords: optimal classification problem, classification problem, evolution algorithm, goose-colony algorithm, jumping frogs algorithm.

Введение. Классификация – мощный научный метод. Задача классификации возникает практически во всех областях знаний при анализе результатов исследований, при проектировании и прогнозировании, при оценке и принятии решений. Часто имея простую формулировку, задача классификации оказывается достаточно сложной и неоднозначной. Более того, иногда при попытках классификации возникают интересные парадоксы, связанные с объединением в один класс принципиально различных объектов.

Решение задачи классификации, как правило, включает значительную долю субъективизма, индивидуальных оценок, нечетких, неформальных выводов. Часто на решение этой задачи влияют приоритеты лица, принимающего решение (ЛПР). Это приводит к построению принципиально различных классификаций на основе одной и той же первичной информации. Особенно ча-

сто эта ситуация возникает в тех областях знаний, в которых невозможно использовать числовые оценки при классификации объектов и явлений, в силу чего возникает необходимость нечетких оценок, использования понятий «схожи», «подобны».

Постановка задачи. Целью настоящей работы является построение метаэвристик для поиска субоптимальной классификации, определенной отношением толерантности на конечном множестве. Такой подход позволяет строить близкие к оптимальным разбиения множества в соответствии с отношением «близости» элементов. Причем это отношение близости не является транзитивным. Предложенные алгоритмы могут найти широкое применение в прикладных задачах, связанных с проблемой классификации объектов по ряду признаков. Такие задачи часто возникают в экономических, социальных и технических науках.

Метод решения и анализ полученных результатов. С точки зрения математики задача классификации может рассматриваться с различных позиций. Основным является теоретико-множественный подход при построении классификации. Однако на практике оказывается, что данный подход хорош лишь post factum, то есть для пояснения и формального описания уже построенной классификации.

Наиболее распространенными до настоящего времени остаются статистические модели классификации, которые позволяют группировать объекты по результатам статистического анализа данных [1,2,3]. Метрические алгоритмы используют формализацию понятия схожести между объектами и гипотезу компактности [2,3,4]. Существует и другой принцип: так называемые логические алгоритмы классификации. В основу этого подхода положен принцип индуктивного вывода логических закономерностей или индукция правил [5,6,7]. Все большее распространение получают модели классификации на основе нечеткой математики, использующие инструментарий теории нечетких множеств. Сравнительно новым направлением являются модели классификации, построенные на основе интегральной математики. Интересным направлением является использование для решения задач классификации методов искусственного интеллекта. Обзор существующих методов распознавания приведен в монографии [6].

Постановка задачи классификации на конечном множестве.

Разбиением конечного множества X будем называть набор его непустых подмножеств $X_1, X_2, ..., X_n$ такой, что

1) $\bigcup_{i=1}^{n} X_i = X$:

2) $\forall i, j, i \neq j \quad X_i \cap X_j = \varnothing, \quad i, j = 1, 2, ..., n$.

Задача классификации на конечном множестве состоит в поиске разбиения, обладающего некоторыми заданными свойствами. Разбиение множества задает каноническое отношение эквивалентности, связанное с этим разбиением. А именно: два элемента считаются эквивалентными, если они принад-

лежат одному элементу разбиения. С другой стороны, легко показать, что любое отношение эквивалентности на конечном множестве определяет его разбиение на классы эквивалентных между собой элементов.

Напомним, что отношением эквивалентности на множестве X называется бинарное отношение "$\sim$", обладающее следующими свойствами:

1) рефлексивность: $\forall x \in X \quad x \sim x$;

2) симметричность: $\forall x, y \in X \quad x \sim y \Rightarrow y \sim x$;

3) транзитивность: $\forall x, y, z \in X \quad x \sim y, y \sim z \Rightarrow x \sim z$.

Приведем простой алгоритм [8], позволяющий по заданному отношению эквивалентности построить соответствующее ему разбиение множества X на классы эквивалентных элементов.

Шаг 0. Выбирается произвольное упорядочивание (нумерация) элементов множества X: $x_1, x_2, ..., x_N \in X$. Здесь $N = |X|$. Определяется множество представителей классов эквивалентности, которое на начальном этапе работы алгоритма пусто. Множество классов эквивалентности также пусто.

….

Шаг i. Выбирается очередной элемент x упорядоченной последовательности элементов множества X и последовательно сравнивается с множеством представителей уже определенных классов эквивалентности. Если этот элемент эквивалентен представителю x_k класса X_k, то он помещается в класс X_k. Если он не эквивалентен ни одному из элементов множества представителей классов, то элемент заносится во множество представителей и определяет новый класс эквивалентности.

Алгоритм заканчивает работу, когда все элементы будут просмотрены и разнесены по классам. Результатом работы алгоритма является множество представителей разных классов и набор классов эквивалентных элементов.

Из транзитивности отношения эквивалентности вытекает, что полученный в результате работы алгоритма набор классов не зависит от первоначального упорядочивания элементов множества X (Шаг 0). Другое упорядочивание может изменить лишь последовательность классов эквивалентности и множество представителей. Описанный выше алгоритм отыскания классов эквивалентности и набора представителей будем называть линейным.

Отметим одну особенность линейного алгоритма. Он может быть применен не только для отношения эквивалентности, но и для любого бинарного отношения. Однако, если отношение не транзитивно, то результат работы алгоритма будет уже существенно зависеть от выбора первоначального упорядочивания элементов.

Отношение толерантности и классификации на основе понятия близости элементов.

Большинство существующих классификаций в прикладных науках строится не на основе отношения эквивалентности, а на основе другого бинарного отношения – отношения толерантности. Отношение толерантности это ре-

флексивное и симметричное отношение "$\approx$" на множестве X, то есть отношение, которое определяется следующими свойствами:

1. $\forall x \in X \quad x \approx x$;
2. $\forall x, y \in X \quad x \approx y \Rightarrow y \approx x$.

Типичным примером подобного отношения является отношение приближенного равенства на множестве чисел. На практике отношение толерантности появляется в виде отношения между объектами, которое описывается словами «подобный», «близкий», «похожий».

Если на конечном множестве X определено отношение толерантности "$\approx$", то можно применить линейный алгоритм выделения классов и получить классификацию на этом множестве. Однако в отличие от классификаций, построенных на основе отношения эквивалентности, классификация, построенная на основе отношения толерантности, существенно зависит от выбора начального упорядочения элементов множества X. Разные способы упорядочивания элементов могут приводить к принципиально различным классификациям.

Критерий оптимальности классификаций.

Существует множество подходов к определению оптимальной классификации. Неформально, классификация является оптимальной, если элементы внутри классов «достаточно близки» друг другу, а сами классы «достаточно удалены» друг от друга. Рассмотрим один из таких подходов.

Мерой близости на конечном множестве X будем называть функцию $p : X \times X \to R_+$, обладающую следующими свойствами:

1) $\forall x, y \in X \quad p(x, y) \geq 0$, причем $p(x, y) = 0 \Leftrightarrow x = y$;

2) $\forall x, y \in X \quad p(x, y) = p(y, x)$.

В частности, мерой близости может служить расстояние между точками в метрическом пространстве.

Пусть задано положительное число $\varepsilon > 0$. Будем говорить, что элементы $x, y \in X$ находятся в отношении близости (близки друг к другу), если $p(x, y) \leq \varepsilon$. Это отношение является отношением толерантности и, как было показано выше, порождает множество различных классификаций, которые определяются выбранным упорядочением на множестве X. Будем называть такие классификации ε-классификациями. Линейный алгоритм построения разбиения немного изменяется. А именно: на шаге с номером i находится класс (среди построенных), представитель которого наиболее близок к анализируемому элементу. Если мера близости между этим представителем и рассматриваемым элементом меньше или равна величине ε, то элемент добавляется в класс. В противном случае рассматриваемый элемент становится представителем нового класса.

Расстояние между двумя непустыми непересекающимися подмножествами $A, B \subseteq X$ определим, как функцию $p(A, B) = \min\limits_{x \in A, y \in B} p(x, y)$.

Пусть задан линейный порядок "$\prec$" элементов на множестве X, определенный некоторой перестановкой $s \in S_n$. Обозначим $X_1, X_2, ..., X_n$ классы эквивалентности для ε-классификации, порождаемой этим порядком. В качестве критерия оптимальности классификации выберем функцию $F(s) = \min\limits_{i,j,i\neq j} p(X_i, X_j)$. Тогда условием оптимальности ε-классификации будет условие

$$F(s) \xrightarrow[s\in S_n]{} \max .$$

Подобная задача является сложной в вычислительном смысле[9]. Однако сам алгоритм поиска разбиения по заданному отношению порядка имеет полиномиальную трудоемкость.

Таким образом, задача поиска оптимальной классификации сводится к задаче поиска оптимальной перестановки элементов множества X. Это позволяет для отыскания субоптимальных решений задачи классификации предложить ряд метаэвристик, построенных для задач имеющих фрагментарную структуру [10]. Приведем краткие описания некоторых из таких метаэвристик.

Эволюционный алгоритм.

Используется стандартная схема эволюционного алгоритма на перестановках. Опишем кратко принцип работы такого алгоритма [10]. В качестве базового множества решений выбирается множество S_n всех перестановок из n элементов. На начальном шаге с помощью оператора начальной популяции строится множество решений $Y_0 \subseteq S_n$. На каждом очередном шаге предполагается заданным некоторое множество перестановок – текущая популяция. На первом шаге это множество $Y = Y_0$. Для каждого из элементов множества Y вычисляется значение критерия селекции, который в рассматриваемом случае является накрывающим отображением исходной задачи. Далее с помощью оператора отбора в текущей популяции Y выбирается множество пар для кроссовера. К каждой паре $U = (u_1, u_2, ..., u_n)$ и $V = (v_1, v_2, ..., v_n)$ из выбранного множества пар применяется оператор кроссовера $Cross(U, V)$, а затем к результату кроссовера применяется оператор мутации. Перестановка – потомок строится следующим образом: последовательности U и V просматриваются с начала. На k-м шаге выбирается наименьший из первых элементов последовательностей и добавляется в новую перестановку – потомок. Затем этот элемент удаляется из двух последовательностей-родителей. Например,

$$Cross((2,4,7,6,1,3,5,8),(5,8,1,3,4,2,6,7)) = (2,4,5,7,6,1,3,8).$$

Оператор мутации M с заданной вероятностью $\alpha \in (0,1)$ выполняет случайную транспозицию (замену местами двух элементов) в перестановке.

Таким путем находится множество элементов – потомков $\tilde{Y}$. К промежуточной популяции $Y \cup \tilde{Y}$, которая является объединением текущей популяции и множества потомков, применяется оператор эволюции, который выделяет на этом множестве новую текущую популяцию. Процесс эволюции по-

вторяется до тех пор, пока не будет выполнено условие остановки эволюционного алгоритма. По найденной перестановке восстанавливается решение исходной задачи.

Муравьиный алгоритм.

Идея муравьиного алгоритма – моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь [12,13]. При своем движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьев находить пути, близкие к оптимальным.

Покажем, как подобный механизм применить к поиску оптимальной перестановки. Процедура вычисления будет состоять из ряда циклов расчета. Каждый путь муравья между позициями 1,2,..,*n* будет определяться перестановкой $s = (i_1, i_2, ..., i_n)$. Муравьи имеют собственную «память». У каждого муравья есть список уже посещенных позиций – список запретов. Обозначим $J_{i,k}^t$ список позиций, которые на цикле *t* необходимо посетить *k*-му муравью, находящемуся в позиции *i*.

Муравьи обладают «обонянием» – они могут улавливать след феромона, подтверждающий желание муравья пройти из позиции *i* в позицию *j* на основании опыта других муравьев. Количество феромона на цикле с номером *t* при переходе из позиции *i* в позицию *j* определяется величиной $\tau_{ij}\ t$. На начальном этапе это количество можно задавать произвольно.

Вероятность перехода *k*-го муравья из позиции *i* в позицию *j* на цикле с номером *t* определяется следующим соотношением:

$$P_{ij,k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}}{\sum\limits_{l \in J_{i,k}^t} [\tau_{il}(t)]^{\alpha}}, & j \in J_{i,k}^t, \\ 0, & j \notin J_{i,k}^t \end{cases}$$

где α – параметр, задающий вес следа феромона. Количество откладываемого феромона составляет величину:

$$\Delta\tau_{ij,k}(t) = \begin{cases} \dfrac{Q}{L_k(t)}, (i,j) \in T_k(t) \\ 0, (i,j) \notin T_k(t) \end{cases},$$

где *Q* – положительный параметр, $L_k(t)$ – значение накрывающего отображения на перестановке, соответствующей маршруту *k*-го муравья на цикле с номером *t*. Изменение количества феромона определяется следующим выражением:

$$\tau_{ij}(t+1) = (1-p)\cdot\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij,k}(t),$$

где m – количество муравьев, p – коэффициент «испарения» ($0 \le p \le 1$).

Алгоритм прекращает работу, когда выполнено некоторое правило остановки, например, достигнута граница числа циклов. Минимальная по значению накрывающего отображения перестановка, найденная на последнем цикле, преобразуется в решение исходной задачи.

Метод прыгающих лягушек.

Алгоритм прост для понимания и реализации, имеет небольшое количество параметров, успешно применялся для решения задач комбинаторной и непрерывной оптимизации [14,15].

Суть алгоритма прыгающих лягушек для поиска оптимальной перестановки сводится к следующей последовательности.

Шаг 1. Инициализировать начальную популяцию лягушек, как множество точек пространства перестановок S_n с метрикой Кендалла.

Шаг 2. Вычислить значение критерия оптимальности для каждой перестановки из начальной популяции.

Шаг 3. Упорядочить решения в порядке убывания значения критерия оптимальности.

Шаг 4. Разделить виртуальных лягушек (решения) на блоки-мемплексы таким образом, что первая в отсортированном списке виртуальная лягушка попадает в первый мемплекс, вторая заносится во второй мемплекс и т.д.

Шаг 5. Так продолжается пока все лягушки не будут распределены в указанное количество мемплексов.

Шаг 6. В каждом мемплексе с номером $k \in \{1,2,...,K\}$ найти лучшее s_{k1} и худшее s_{k2} решение.

Шаг 7. Попытаться улучшить положение худшей виртуальной лягушки путем случайного перемещения ее в направлении лучшей лягушки. Это происходит применения оператора кроссовера $s = Cross(s_{k2}, s_{k1})$.

Шаг 8. Если предыдущая операция не улучшает решение, то попытаться улучшить положение худшей виртуальной лягушки путем перемещения ее в направлении глобально лучшей лягушки $s = Cross(s_{k2}, s_{k1})$.

Шаг 9. Если и последняя операция не приводит к улучшению позиции виртуальной лягушки, то взамен ее случайным образом создать в области поиска новую лягушку – перестановку.

Шаг 10. Объединить виртуальных лягушек всех мемплексов в одну группу.

Шаг 11. Если условия завершения алгоритма не выполнены, то – переход к Шагу 3.

Шаг 12. Последняя глобально лучшая виртуальная лягушка соответствует субоптимальному решению задачи.

Выводы. В данной статье были рассмотрен метод отыскания оптимальных ε-классификаций на основе известных метаэвристик. Этот подход практически без изменений может быть перенесен и на другие виды классификаций,

которые строятся на основе понятия близости элементов. Точно также кроме предложенных в работе метаэвристик могут быть рассмотрены и любые другие метаэвристики, которые применимы к задачам оптимизации на фрагментарных структурах [10].

Библиографические ссылки

1. **Айвазян, С.А.** Прикладная статистика: Классификация и снижение размерности. Справочное издание [Текст] / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин. – М., 1989. – 607 с.
2. **Дюран, Б.** Кластерный анализ [Текст] / Б. Дюран, П. Оделл. – М., 1977. – 128 с.
3. **Ким, Дж.** Факторный, дискриминантный и кластерный анализ [Текст] / Дж. Ким, Ч.У. Мьюлер, У.Р. Клекка. – М., 1989. – 216 с.
4. **Вайнцвайг, М.Н.** Алгоритм обучения распознаванию образов «кора» [Текст] / М.Н. Вайнцвайг // Алгоритмы обучения распознавания образов. – 1973. – С. 110–116.
5. **Дюличева, Ю.Ю.** Стратегия редукции решающих деревьев (обзор) [Текст] / Ю.Ю. Дюличева // Таврический вестник информатики и математики. – 2002. – №1. – С. 10–17.
6. **Журавлев, Ю.И.** Распознавание. Математические методы. Программная система. Практические применения [Текст] / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько. – М., 2006. – 176 с.
7. **Лбов, Г.С.** Методы обработки разнотипных экспериментальных данных [Текст] / Г.С. Лбов. – Новосибирск, 1981. – 160 с.
8. **Перепелица, В.** Задачи классификации и формирование знаний [Текст] / В. Перепелица, И. Козин, Э. Терещенко. – Germany, 2012. – 196 с.
9. **Гэри М.** Вычислительные машины и труднорешаемые задачи [Текст] / М. Гэри, Д. Джонсон; пер. А. Фридман. – М., 1982. – 416 с.
10. **Kozin, I.V.** Fragmentary Structures in Discrete Optimization Problems [Text] / I.V. Kozin, N.K. Maksyshko, V. A. Perepelitsa // Cybernetics and Systems Analysis. – 2017, Volume 53, Issue 6. – P. 931–936. – DOI: https://doi.org/10.1007/s10559-017-9995-6
11. **Козин, И.В.** Эволюционный алгоритм оптимальной классификации [Текст] / И.В. Козин // Искусственный интеллект. – 2015. – № 3-4 (69-70). – С. 98–104.
12. **Dorigo, M.** Optimization, Learning, and Natural Algorithms [Текст] / M. Dorigo. – PhD Thesis, Dipartimento di Elettronica, Politechnico. – Di Milano, Italy. 1992. – 140 p.
13. **Штовба, С.Д.** Муравьиные алгоритмы: теория и применение. Программирование [Текст] / С.Д. Штовба. – М., 2005. – 230 с.
14. **Карпенко, А.П.** Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой : учебное пособие для вузов [Текст] / А.П. Карпенко. – М., 2014. – 446 с.
15. **Narimani, M.R.** A New Modified Shuffle Frog Leaping Algorithm for NonSmooth Economic Dispath [Текст] / M.R. Narimani // World Applied Sciences Journal. 2011. – P. 803–814.

Поступила в редколлегию 16.10. 2020.