УДК 004.023+ 004.855.5

КОМБИНИРОВАНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ И АНСАМБЛИРОВАНИЯ ДЛЯ ПОСТРОЕНИЯ КЛАССИФИКАЦИОННЫХ МОДЕЛЕЙ

В.М.Басинский
Аспирант ГрГУ,
преподаватель

Гродненский государственный университет имени Янки Купалы, Республика Беларусь
E-mail: vadim.basinskii@gmail.com

В.М.Басинский

Окончил Гродненский государственный университет имени Я. Купалы. Работает в ГрГУ в должности преподавателя. Проводит научные исследования в области BigData и применении метаэвристических алгоритмов для решения задач

Аннотация. Описаны принципы сведения задачи классификации к задаче поиска путей на графе, описан примененный метод проведения дискретизации непрерывных атрибутов, проведен анализ метрики, использованной для оценки качества классификаторов, рассмотрен вариант применения генетического алгоритма для обеспечения достаточного для получения качественной классификации разнообразия классификаторов, предложены пути дальнейшего исследования в рамках задачи.

Ключевые слова: классификация, оптимизация, алгоритм муравьиной колонии, генетический алгоритм.

*Введение***.** Формальная постановка задачи классификации может быть представлена следующим образом. Задано множество объектов XX, множество допустимых ответов YY , и существует целевая функция (target function) $y^*{:}y^*{:}\, X \to YX \to Y$, значения которой $y_i y_i$ = $y^*(x_i)y^*(x_i)$ известны только на конечном подмножестве объектов$\{x_1, \dots x_l\} \in X\{x_1, \dots x_l\} \in X$. Пары «объект– ответ» ($x_i x_i$, $y_i\, y_i$) называются прецедентами. Совокупность пар $X^l X^l$ = $(x_i, y_i)\,{}_{i=1}^{l}(x_i, y_i)\,{}_{i=1}^{l}$ называется обучающей выборкой (training sample) [1].

Задача обучения по прецедентам заключается в том, чтобы по выборке $X^l X^l$ восстановить зависимость $y^* y^*$, то есть построить решающую функцию (decision function) $f{:}X \to Yf{:}X \to Y$, которая приближала бы целевую функцию $y^*(x)y^*(x)$, причём не только на объектах обучающей выборки, но и на всём множестве XX.

Каждый объект описывается в виде набора выражений $term_{ij} = (A_i = b_j)$ $term_{ij} = (A_i = b_j)$, где $A_i A_i$ – атрибут, а $b_j b_j$ – одно из значений домена этого атрибута.

Задачу классификации можно представить в виде графа следующим образом:

1. Часть вершин графа соответствует набору выражений – эти вершины позволяют описать объекты выборки.

2. Часть вершин графа соответствует элементам множества – эти вершины позволяют описать допустимые классы.

3. Каждая вершина связана со всеми остальными, кроме вершин которые соответствуют другим значениям этого же атрибута.

4. Граф является неориентированным – переход можно осуществить между любой допустимой парой вершин.

После такого преобразования задача классификации может быть представлена в схожем виде как показано на рисунке 1.

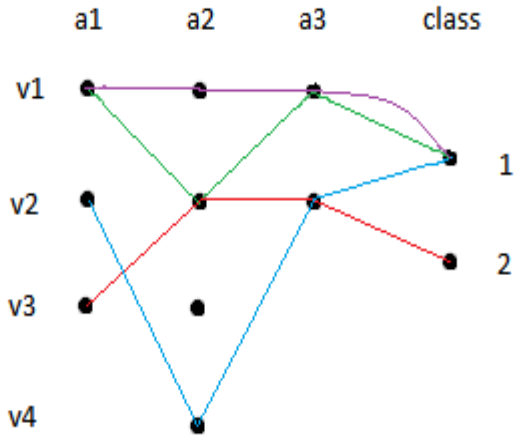

Рисунок *1.* – Представление задачи классификации в виде графа

Каждая допустимая запись из тренировочной выборки представляет собой один путь на таком графе. На рисунке 1 показано четыре пути, которые соответствуют следующим четырем записям:

1. $a_1 = v_1 \,\&\&\, a_2 = v_1 a_1 = v_1 \,\&\&\, a_2 = v_1 \,\&\&\, a_3 = v_1 \rightarrow class = 1 a_3 = v_1 \rightarrow class = 1,$
2. $a_1 = v_1 \,\&\&\, a_2 = v_2 a_1 = v_1 \,\&\&\, a_2 = v_2 \,\&\&\, a_3 = v_1 \rightarrow class = 1 a_3 = v_1 \rightarrow class = 1,$
3. $a_1 = v_2 \,\&\&\, a_2 = v_4 a_1 = v_2 \,\&\&\, a_2 = v_4 \,\&\&\, a_3 = v_2 \rightarrow class = 1 a_3 = v_2 \rightarrow class = 1,$
4. $a_1 = v_3 \,\&\&\, a_2 = v_2 a_1 = v_3 \,\&\&\, a_2 = v_2 \,\&\&\, a_3 = v_2 \rightarrow class = 2 a_3 = v_2 \rightarrow class = 2.$

Результатом работы классификатора будет являться набор *классификационных правил*, извлеченных из тренировочной выборки.

Классификационным правилом будем называть такой путь в графе, который позволит компактно описать целый набор путей из обучающей выборки и выделить для них общие вершины.

Примером классификационного правила для описанного случая может являться следующее выражение:

$$r_1: a_1 = v_1 \,\&\&\, a_3 = v_1 \rightarrow class = 1$$

Графически это проиллюстрировано на рисунке 2.

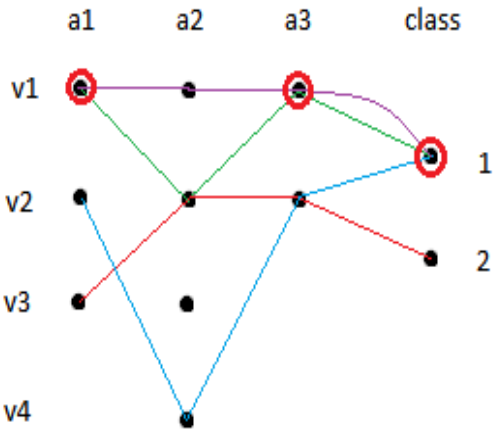

Рисунок *2.* – Пример классификационного правила

Классификационное правило покрывает пути 1 и 2, которые соответствуют первым двум объектам тренировочной выборки, и, с точки зрения задачи классификации, может использоваться для классификации этих объектов.

Большое влияние на сложность решения задачи классификации оказывает структура входных данных. Если атрибут имеет большое количество значений, то растет число вершин в графе для соответствующего атрибута, а среднее число путей через каждую из них уменьшается. Это значительно ухудшает качество классификации и точность извлекаемого набора правил. Для решения этой проблемы используется дискретизация атрибутов.

Для проведения дискретизации используется *метод разбиения на подмножества с минимальной энтропией*. Метод использует информационную энтропию возможных разбиений для определения границы разбиения множества на два подмножества.

Пусть имеется набор записей $S\,S$, атрибут AA и граница разбиения TT. Тогда энтропия разбиения множества SS на множества S_1S_1 и S_2S_2может быть определена следующим образом:

$$E(A,T,S) = \frac{|S_1|}{|S|} * E(S_1) + \frac{|S_2|}{|S|} * E(S_2)E(A,T,S) = \frac{|S_1|}{|S|} * E(S_1) + \frac{|S_2|}{|S|} * E(S_2).$$

Разбиение продолжается на наборах S_1S_1 и S_2S_2 пока не будет выполняться условие остановки, описанное выражением:

$$Gain(A,T,S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T,S)}{N}Gain(A,T,S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T,S)}{N}.$$

Величина $Gain(A,T,S)\,Gain(A,T,S)$ определяется как разность начального и полученного разбиений, а величина $\Delta(A,T,S)\Delta(A,T,S)$ определяется по следующей формуле:

$$\Delta(A,T,S) = \log_2(3^k - 2) - (k * E(S) - k_1 * E(S_1) - k_2 * E(S_2))$$
$$\Delta(A,T,S) = \log_2(3^k - 2) - (k * E(S) - k_1 * E(S_1) - k_2 * E(S_2)).$$

Оценка качества построенных классификационных правил производится на основании расчета меры покрытия (*coverage*) и меры уверенности (*confidence*). Определяются они так следующим образом:

$$coverage = \frac{|T_{ij}\ \&\ class{=}True|}{|T_{ij}|} coverage = \frac{|T_{ij}\ \&\ class{=}True|}{|T_{ij}|},$$

$$confidence = \frac{|covered\ by\ rule\ \&\ class{=}True|}{|covered\ by\ rule|}$$

$$confidence = \frac{|covered\ by\ rule\ \&\ class{=}True|}{|covered\ by\ rule|},$$

где $T_{ij}\,T_{ij}$ – набор вершин, которые рассматриваемое правило покрывает.

В этом случае показатель качества правила будет рассчитываться по формуле:

$$Q = coverage + confidence$$

Несмотря на наличие других метрик оценки качества классификационного правила, использование именно такой схемы выглядит наиболее интуитивно предпочтительным и понятным, так как метрика основана именно на плотности путей через некоторый набор вершин графа.

Построение и поиск классификационных правил на графе будет осуществлять алгоритм муравьиной колонии [3]. Алгоритм муравьиной колонии имеет набор параметров [4-6], значения которых необходимо подбирать под каждый набор индивидуально, так как на основе проведения экспериментов и анализа полученных результатов было установлено, что их оптимальные значения зависят от структуры и характеристик данных в выборке.

Зависимость значений параметров от структуры данных в явном виде аналитическими методами получить невозможно, а осуществлять подбор параметров вручную представляется затруднительным, поэтому предлагается реализовать подбор и оптимизацию параметров при помощи генетического алгоритма.

С учетом того, что обычно использование ансамблей классификационных моделей позволяет увеличить качество классификации, было решено применить схожий подход [2].

Каждая муравьиная колония является одной классификационной моделью. Возможность получения разнообразных моделей обеспечивается двумя элементами: генетическим алгоритмом для модификации параметров колоний и небольшими вариациями наборов атрибутов (т.е. подграфов), по которым колония будет строить пути. Идея разбиения множества атрибутов на некоторые подмножества взята из алгоритма леса случайных деревьев, но проведенное тестирование показало, что предварительное выполнение факторного анализа атрибутов для выделения подмножеств, обеспечивает лучшее качество, чем случайный выбор подмножеств.

Отсутствие обмена информацией и опытом между муравейниками не является критичной проблемой само по себе, но благодаря использованию генетического алгоритма для управления ходом работы алгоритма муравьиной колонии, можно считать, что эта проблема решается при проведении скрещивания и генерации колоний, следующих после начального поколений.

В общем виде процесс обучения классификаторов можно сформулировать следующим образом: генетический алгоритм будет работать с коллекциями муравьиных колоний, которые будут генерироваться поколениями на основании результатов работы предыдущих итераций.

Описанная модель позволяет объединять преимущества как методов ансамблирования, позволяющих разбивать задачу на множество подзадач и распараллеливать процесс поиска решения, так и эволюционных алгоритмов случайного поиска для получения разнообразия классификационных моделей и направленного процесса оптимизации гиперпараметров.

Предлагаемыми модификациями описываемого алгоритма являются следующие подходы:

1. Изменение принципа построения графа – отказ от построения полносвязного графа в пользу выделения нескольких компонент сильной связности по подмножествам атрибутов, связанных с главными компонентами из факторного анализа. В этом случае предлагается разрешать случайный переход между компонентами графа с вероятностью пропорциональной корреляциям атрибутов и факторов.
2. Использование весов графа в качестве инициализатора весов нейронной сети и отказ от применения для классификации подходов сходных с деревьями принятия решений.

Заключение. На основании приведенных рассуждений планируется реализация программного модуля, позволяющего осуществлять решение задач классификации при помощи комбинирования метаэвристических и эволюционных алгоритмов для построения классификационных моделей с методами ансамблирования или нейронными сетями в качестве классификатора.

Список литературы

[1.] Witten, I.H. Data Mining: Practical Machine Learning Tools and Techniques, Third Edition / I.H. Witten, E. Frank, M.A. Hall. – Morgan Kaufmann Series in Data Management Systems, 2012.

[2.] Cinaroglu, S. Comparison of Performance of Decision Tree Algorithms and Random Forest: An Application on OECD Countries Health Expenditures International Journal of Computer Applications (0975 – 8887) Volume 138 – No.1, March 2016

[3.] Parpinelli, R.S. Data mining with an ant colony optimization algorithm. / R.S. Parpinelli, H.S. Lopes, A.A. Freitas. – IEEE Transactions on Evolutionary Computation, Special Issue on Ant Colony Algorithms, 6(4), 321-332.

[4.] Liu B. Classification rule discovery with Ant Colony Optimization / B. Liu, H.A. Abbass, B. McKay. – IEEE Computational Intelligence Bulletin Vol.3 No.1

[5.] Ants Constructing Rule-Based Classifiers / Martens D. [и др.] – Swarm Intelligence in Data Mining, 2006.

[6.] Zhang X. An Adaptive Ant Colony Algorithm for Classification Rule Mining / X. Zhang, W. Sun. – Advances in Intelligent Systems Research, volume 133

COMBINING THE GENETIC ALGORITHM, ANT COLONY ALGORITHM AND ENSEMBLING FOR SOLVING THE CLASSIFICATION PROBLEM

V.M. BASINSKII
Postgraduate student of the GrSU, lecturer GrSU

Yanka Kupala Grodno State University, Republic of Belarus
E-mail: vadim.basinskii@gmail.com

Abstract. The principles of changing the classification problem to the equal problem of finding paths on a graph are described, the applied method for discretizing continuous attributes is shown, the metrics used to evaluate the quality of classifiers are analyzed, the application of a genetic algorithm to ensure sufficient variety of classifiers to obtain a high-quality classification is considered, the ways of further research in the scope of the task are proposed.

Keywords: classification, optimization, ant colony algorithm, genetic algorithm.