УДК 519.68

САМОКОНФИГУРИРУЕМЫЙ АВТОКАТАЛИТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КЛАССИФИКАЦИИ

Семенкина О.Е.
научный руководитель д-р физ.-мат. наук Попов Е.А.
Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева

Одной из наиболее активно развивающихся областей современной науки является интеллектуальный анализ данных, применение которого на практике широко распространено. Одним из наиболее используемых методов анализа данных является классификация, состоящая в сопоставлении каждому объекту определенного класса, основываясь на предыдущем опыте. Классификация также имеет широкое применение на практике, например, в скоринге, категоризации, распознавании речи, диагностике и так далее.

Многие задачи классификации достаточно трудны, и стандартные методы могут не справиться с их решением, поэтому для их решения зачастую применяются стохастические методы. Данная статья посвящена методу классификации Ant Miner, который основан на базе автокаталитического алгоритма комбинаторной оптимизации Ant Colony Optimization (ACO) [1] и позволяет строить эффективную базу правил классификаторов. Алгоритм Ant Miner впервые был описан в [2], а позднее, например в [3], были введены параметры α - относительная важность эвристики и β - относительная важность феромона, по аналогии с классическим ACO.

В предыдущих публикациях автором исследовались муравьиный алгоритм (ACO), генетический алгоритм (Genetic Algorithm (GA или ГА)) [4], алгоритм умных капель (Intelligent Water Drops (IWDs)) [5] и эвристика Лина-Кернигана [6] применительно к задаче коммивояжера [7], а также проводилось сравнение их эффективности с самоконфигурируемыми алгоритмами [8], позволяющими выбирать параметры алгоритмов в ходе решения задачи. Результаты показали, что используемый метод адаптации [9] показывает высокую эффективность, а значит модификация и других алгоритмов таким же образом может оказать положительное влияние на их работу. В данной стать исследуется самоконфигурируемый алгоритм классификации на основе муравьиного алгоритма - Self-configuring Ant Miner (SCAM) – эффективность которого в большинстве случаев оказывается выше, чем у исходного алгоритма Ant Miner.

В своей работе алгоритм Ant Miner проектирует классификатор, который состоит из правил следующего вида:

IF <*conditions*> THEN < *class*>,

где <*conditions*> состоит из некоторого множества термов и определяет значения атрибутов, то есть имеет вид

IF $term_1$ AND $term_2$ AND ...,

а часть < *class*> определяет принадлежность к тому или иному классу.

Каждый терм является тройкой вида:

<*attribute, operator, value*>,

где *attribute* - наименование (или номер) некоторого атрибута, фиксируемого в этом терме, *operator* - это способ сравнения двух значений атрибутов (например, знак равенства "="), а

value - конкретное фиксируемое значение выбранного атрибута из его множества значений.

В начале своей работы алгоритм принимает всю обучающую выборку и, основываясь на ней, строит правило классификации следующим образом:

1. Инициализируется начальное количество феромона на каждом из возможных термов;
2. Пока последовательность правил не сойдется:
   2.1. *i*-ый муравей строит правило классификации;
   2.2. Правило оптимизируется посредством удаления термов или изменения класса, если при этом эффективность правила увеличивается;
   2.3. Обновляется количество феромона на использованных в этом правиле термах;
3. Из всех рассмотренных правил выбирается лучшее и записывается в список открытых правил;
4. Из обучающей выборки удаляются все те случаи, которые покрываются найденным правилом;

Описанная последовательность шагов повторяется до тех пор, пока в обучающей выборке не останется количество случаев, меньшее, чем фиксированное число N_{uc}.

Серьезным недостатком стохастических алгоритмов является большое количество настраиваемых параметров, влияющих на эффективность работы алгоритма. Зависимость эффективности алгоритма от его настроек не очевидна, потому их настройка зачастую производится произвольно или, в лучшем случае, после статистических исследований. Это связано с тем, что даже для экспертов в данной области выбор настроек является сложной задачей, а значит автоматическая настройка параметров алгоритма во время его работы представляется перспективной идеей.

Общая суть подхода к выбору эффективных вариантов настроек алгоритма, примененного для создания самоконфигурируемого муравьиного алгоритма для задач классификации, состоит в следующем [9]. Для каждого параметра алгоритма выбор его значения осуществляется отдельно, z – количество рассматриваемых значений данного параметра. В начале своей работы алгоритм считает все значения параметра равновероятными, а через некоторое время производится оценка эффективности каждого значения параметра следующим образом:

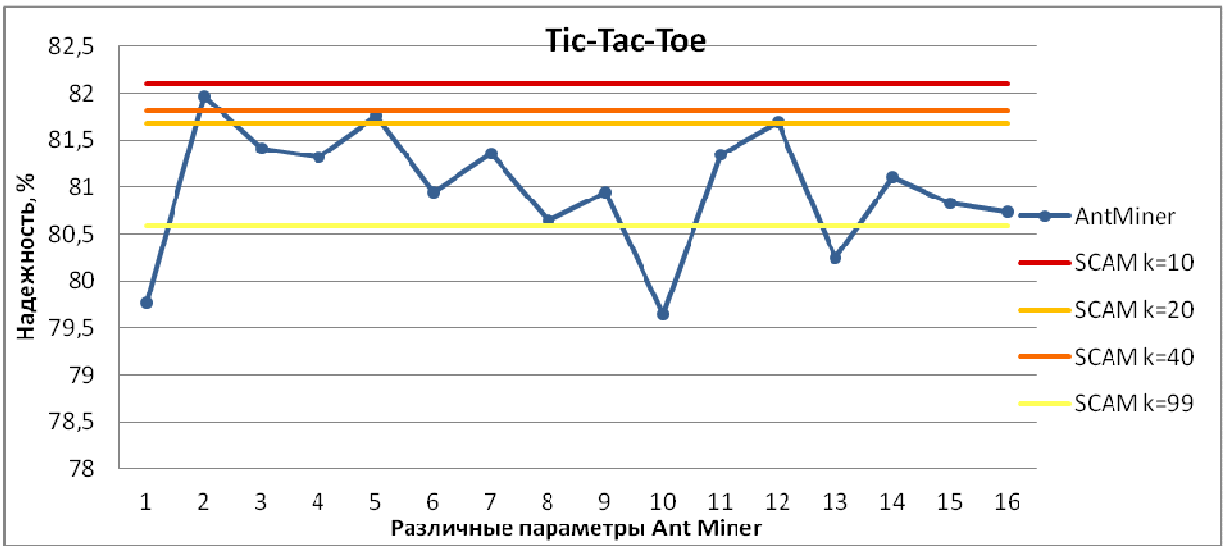
$$averagefit\,ness_i = \frac{\sum_{j=1}^{n_i} f_{ij}}{n_i}, \qquad i = 1,2,\ldots,z,$$

где n_i – количество муравьев, использовавших при построении решения i-е значение параметра; f_{ij} –оценка качества решения, построенного j-м муравьем с использованием i-го значения параметра; $averagefitness_i$ – средняя пригодность решений, построенных с использованием i-го значения параметра. Вероятность наиболее эффективного значения параметра увеличивается на число $((z-1)\cdot K)/(z\cdot N)$, а вероятности остальных вариантов уменьшаются на $K/(z\cdot N)$), где N – количество произведенных оценок эффективности параметров с начала работы алгоритма, K – константа. В алгоритме Ant Miner муравьи строят решения последовательно, а не параллельно, поэтому оценка эффективности значений параметров основывается на последних k муравьях, построивших решение последовательно.

Сравнение алгоритма Ant Miner и SCAM проводилось на нескольких задачах, взятых из международного репозитория задач машинного обучения, таких как Tic-Tac-Toe, Dermatology, Ljubljana Breast Cancer, Breast Cancer Wisconsin. Алгоритм Ant Miner

рассматривался с различными вариантами настроек: α = 1, 2, 5, 10, β = 1, 2, 5, 10. Всем алгоритмам давались одинаковые начальные условия, а именно количество муравьев 100, минимальное количество случаев, покрываемых правилом 10, число правил для проверки сходимости 10. Результат работы оценивался по 50 прогонам по таким параметрам, как лучшая эффективность (процент правильно классифицированных примеров тестовой выборки) по всем прогонам алгоритма, средняя эффективность и среднеквадратическое отклонение.

Анализ результатов показывает что в среднем SCAM на большинстве задач показывает более высокую эффективность по сравнению с Ant Miner при средних его настройках, а в некоторых случаях даже при лучших его настройках. На рис. 1 приведено сравнение эффективности алгоритмов Ant Miner и SCAM при различных *k* на примере задачи Tic-Tac-Toe.



Рису. 1. - Сравнение эффективности Ant Miner и SCAM при различных *k*

К тому же существенным недостатком стохастических алгоритмов является большое количество настраиваемых параметров, так как в условиях ограниченности ресурсов может просто не быть возможности перебрать все варианты настроек, а узнать лучшие настройки на конкретной задаче заранее невозможно. Поэтому создание адаптивных алгоритмов, подстраивающихся под конкретную задачу, существенно облегчает их применение на практике и для специалистов, и для обычных пользователей. При добавлении дополнительных вариантов значений параметров работа по выявлению лучших настроек значительно увеличивается, в то время как для самоконфигурируемого алгоритма это не является проблемой. Таким образом, автоматическая настройка параметров в ходе решения задачи является большим преимуществом SCAM, одновремнно с этим в большенстве случаев не ухудшая среднюю его эффективность, а в ряде случаев к тому же повышая ее.

Литература

1. Dorigo M., Gambardella L. M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem // IEEE Transactions on Evolutionary Computation. - 1997. - P. 53–66.
2. Parpinelli R., Lopes H., Freitas A. Data Mining with an Ant Colony Optimization Algorithm // IEEE Transactions on Evolutionary Computation 6(4). - 2002. - P. 321 - 332.
3. Sameh A., Magdy K. Data Mining Ant Colony for Classifiers // International Journal of Basic & Applied Sciences. - vol. 10. - Issue 3. - 2010. - P. 28 - 35.
4. Eiben A.E., Smith J.E. Introduction to Evolutionary Computing, Springer, 2003. ISBN:3540401849
5. Shah-Hosseini H. Optimization with the Nature-Inspired Intelligent Water Drops Algorithm // Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, 2009.
6. Lin S. Computer Solutions of the Traveling Salesman Problem // BSTJ. - v.44. - No. 10. - 1965. - P. 2245-2269.
7. Семенкина О.Е., Семенкина О.Э. Исследование эффективности бионических алгоритмов комбинаторной оптимизации // Программные продукты и системы. - № 3 (103). - 2013. - С. 129-133.
8. *Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G.* Parallel Problem Solving from Nature // PPSN XI 11th International Conference, Kraków, Poland, September 11-15, 2010.
9. Semenkin, E., Semenkina, M. Self-Configuring Genetic Algorithm with Modified Uniform Crossover Operator // Tan, Y., Shi, Y., Ji, Z. (Eds.): Advances in Swarm Intelligence, ICSI 2012, Part 1, LNCS 7331, Springer, Heidelberg, 2012, P. 414-421.