УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
май—июнь 2019 Том 19 № 3 ISSN 2226-1494 http://ntv.itmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTCS
May—June 2019 Vol. 19 No 3 ISSN 2226-1494 http://ntv.itmo.ru/en/
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

УДК 004.021:004.852:004.832.23 doi: 10.17586/2226-1494-2019-19-3-508-515

АВТОМАТИЧЕСКАЯ НАСТРОЙКА ГИПЕРПАРАМЕТРОВ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ С ПОМОЩЬЮ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

С.Б. Муравьёв, В.А. Ефимова, В.В. Шаламов, А.А. Фильченков, И.Б. Сметанников

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: mursmail@gmail.com

Информация о статье
Поступила в редакцию 18.02.19, принята к печати 15.04.19
Язык статьи — русский

Ссылка для цитирования: Муравьёв С.Б., Ефимова В.А., Шаламов В.В., Фильченков А.А., Сметанников И.Б. Автоматическая настройка гиперпараметров алгоритмов кластеризации с помощью обучения с подкреплением // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 3. С. 508–515. doi: 10.17586/2226-1494-2019-19-3-508-515

Аннотация

Предмет исследования. Исследованы алгоритмы выбора и настройки модели алгоритма в задачах кластеризации, применяемые в машинном обучении. Подробно рассмотрен метод выбора модели, показана необходимость поиска компромисса между исследованием и эксплуатацией, который производится с помощью сведения задачи к задаче о многоруком бандите. **Метод.** В работе предложен алгоритм одновременного выбора модели и настройки ее гиперпараметров на основе сведения к задаче о многоруком бандите. Предложены вариации алгоритма, использующие различные способы решения задачи о многоруком бандите, Softmax и UCB1, кроме того, награда определялась разными способами. **Основные результаты.** Проведенные эксперименты на реальных наборах данных из репозитория UCI позволили подтвердить, что предложенный алгоритм в целом за фиксированное время достигает существенно лучших результатов, чем метод полного перебора, а также позволили определить наиболее успешную вариацию предложенного алгоритма. **Практическая значимость.** Предложенный алгоритм может быть использован для выбора и настройки модели алгоритма кластеризации, в нем может использоваться любой алгоритм оптимизации гиперпараметров. Соответственно он может быть применен в широком спектре задач кластеризации, например, в биологии, психологии и при обработке изображений.

Ключевые слова
машинное обучение, кластеризация, настройка гиперпараметров, обучение с подкреплением, многорукий бандит

Благодарности
Работа выполнена при финансовой поддержке Правительства Российской Федерации, субсидия 08-08 и РФФИ, грант 16-37-60115-мол_а_дк.

doi: 10.17586/2226-1494-2019-19-3-508-515

AUTOMATIC HYPERPARAMETER OPTIMIZATION FOR CLUSTERING ALGORITHMS WITH REINFORCEMENT LEARNING

S.B. Muravyov, V.A. Efimova, V.V. Shalamov, A.A. Filchenkov, I.B. Smetannikov

ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: mursmail@gmail.com

Article info
Received 18.02.19, accepted 15.04.19
Article in Russian

For citation: Muravyov S.B., Efimova V.A., Shalamov V.V., Filchenkov A.A., Smetannikov I.B. Automatic hyperparameter optimization for clustering algorithms with reinforcement learning. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 3, pp. 508–515 (in Russian). doi: 10.17586/2226-1494-2019-19-3-508-515

Abstract

Subject of Research. The paper deals with research of clustering algorithms for hyperparameters optimization used in machine learning. Model selection problem is comprehensively studied, and the need of the tradeoff between exploration and exploitation is identified. Thus, the problem is reduced to multi-armed bandit problem. **Method.** The paper presented the approach for simultaneous algorithm selection and hyperparameters optimization. We used solution of the Multiarmed Bandit problem and considered Softmax- and UCB1-based algorithm variants in combination with different reward functions. **Main Results.** Experiments on various datasets from UCI repository were carried out. The results of experiments confirmed that proposed

algorithms in general achieve significantly better results than exhaustive search method. It also helped to determine the most promising version of the algorithm we propose. **Practical Relevance.** The suggested algorithm can be successfully used for model selection and configuration for clustering algorithms, and can be applied in a wide range of clustering tasks in various areas, including biology, psychology, and image analysis.

Keywords
machine learning, clustering, algorithm selection, hyperparameter optimization, multi-armed bandit, reinforcement learning

Acknowledgements
This work was financially supported by the Government of Russian Federation, grant 08-08 and the Russian Foundation for Basic Research, Grant 16-37-60115 mol_a_dk.

Введение

Задача кластеризации заключается в разбиении множества объектов на подмножества похожих между собой объектов. Кластеризация нацелена на обнаружение естественных структур в данных [1–4]. Задача кластеризации возникает во многих областях, среди которых распознавание образов [3], биология [4], психология [5], обработка изображений [6], компьютерная безопасность [7] и др.

В случае применения кластеризации на практике нужно оценивать полученные кластеры с точки зрения исследуемой предметной области и определять, передают ли они истинные связи между объектами [8]. Считается, что не существует четкого и однозначного определения кластера [9, 10]. Было предложено множество методов анализа кластеров, но в большинстве таких исследований авторы не определяют четко, что же их метод должен обнаруживать, и не дают формального определения истинных кластеров [11].

Как и другие алгоритмы машинного обучения, алгоритмы кластеризации при разной конфигурации выдают разный результат на одном наборе данных. Важно, что качество этого результата очень сильно зависит от выбранной конфигурации, т.е. гиперпараметров алгоритма кластеризации.

На практике выбор алгоритма и настройка его гиперпараметров часто выполняется экспертами вручную, что требует значительных временных затрат. Ранее были доступны только компьютеры малых вычислительных мощностей, что оправдывало этот подход. Сейчас такой проблемы нет, что обусловливает актуальность задачи автоматизации этого процесса, поскольку она позволяет существенно сэкономить человеческие усилия за счет использования вычислительных мощностей компьютера, что в свою очередь позволит экспертам тратить время на решение более сложных и важных задач.

Цель данного исследования — разработка и реализация алгоритма, который позволит за фиксированный промежуток времени автоматически находить наиболее подходящий алгоритм кластеризации и оптимизировать его гиперпараметры для конкретного набора данных. Требуется разработать подход, который позволил бы эффективно производить выбор модели алгоритма кластеризации и настройку ее гиперпараметров.

Автоматический выбор модели алгоритма кластеризации и ее гиперпараметров

Сначала формально определим задачу. Моделью A будем называть алгоритм кластеризации. Каждая модель задается некоторым набором гиперпараметров размерности n обычно специфичным для этой модели, лежащем в пространстве гиперпараметров Λ этой модели, $\lambda = \{\lambda_1, \lambda_2, ...\lambda_n\} \in \Lambda$. Например, гиперпараметром является число кластеров в алгоритме k-Means. Модель с конкретным набором гиперпараметров λ будем обозначать как A_λ.

Пусть задана некоторая мера качества кластеризации Q. Задача оптимизации гиперпараметров заключается в подборе таких $\lambda^* \in \Lambda$, при которых заданная модель алгоритма кластеризации A достигнет наилучшего качества на наборе объектов X: $Q(A_\lambda, X) \to \max_{\lambda \in \Lambda}$. Отметим, что даже один алгоритм с разными гиперпараметрами по-разному разбивает на кластеры множество объектов, что, следовательно, дает разные оценки качества кластеризации. Однако, если оценить каждое из этих разбиений мерами качества, не найдется разбиения, которое будет превосходить все другие по оценкам всех мер [12]. Соответственно возникает задача выбора модели алгоритма кластеризации и оптимизации гиперпараметров выбранной модели.

Задача выбора модели алгоритма и ее гиперпараметров в общем случае формально записывается, как поиск модели алгоритма A_λ^* который бы *минимизировал* Q:

$$A_{\lambda^*}^* \in \arg\min_{\Lambda^j \in \mathbf{A} \in \Lambda^j} Q(A_\lambda^j, D),$$

где $A = \{A^1, A^2, ... A^k\}$ — набор моделей алгоритмов, с каждой из которых связано пространство гиперпараметров $\Lambda^1, \Lambda^2, \ldots, \Lambda^k$ соответственно.

Рассмотрим существующие способы решения задачи выбора модели. Одним из них является мета-обучение. Данный подход включает в себя построение мета-классификатора [13], который по набору данных может рекомендовать один или несколько алгоритмов с конкретными гиперпараметрами из заранее зафиксированного списка. Понятно, что это далеко не всегда приводит к нахождению оптимального решения.

Выбору числа кластеров посвящены многие работы как в общем случае [14, 15], так и для конкретного алгоритма [16]. Число кластеров выбирается в соответствии с характеристиками набора данных или же разбирается результат работы алгоритма. Статей по настройке других гиперпараметров не найдено. Также были изучены современные исследования по теме кластеризации, но автоматический выбор модели кластеризации и оптимизация ее гиперпараметров нигде не описан.

Заметим, что в задаче классификации для выбора модели и настройки ее гиперпараметров было создано несколько подходов: Auto-WEKA [17], Tree-based Pipeline Optimization Tool (TPOT) [18], auto-sklearn [19], в том числе это происходит и методами обучения с подкреплением [20]. Кроме того, библиотека Auto-WEKA 2.0 поддерживает выбор модели в задаче регрессии [21].

Предложенные методы автоматического выбора и настройки гиперпараметров

В данной работе мы рассматриваем задачу автоматического и одновременного выбора модели и ее гиперпараметров, ключевым ресурсом в ней является время оптимизации гиперпараметров каждой модели. Предложенный подход в целом не налагает ограничений на алгоритм настройки гиперпараметров, так что для конкретной модели будем использовать алгоритм случайного поиска и алгоритм SMAC [22]. Некоторые из вариаций предложенного решения используют структуру леса случайных деревьев, на основе которого работает алгоритм SMAC.

В качестве базового метода возьмем *метод полного перебора*: получив на вход набор данных, меру качества (целевую функцию) и временной бюджет, система делит последний поровну между всеми имеющимися алгоритмами и последовательно оптимизирует гиперпараметры относительно данной целевой функции у каждого алгоритма. Когда заканчивается бюджет времени у всех алгоритмов, система сравнивает достигнутые значения целевой функции и выбирает среди настроенных алгоритмов лучший (рис. 1).

Данное решение является довольно прямолинейным и не требует значительных изысканий, однако, оно не было предложено ранее. Рассмотрим более сложный алгоритм.

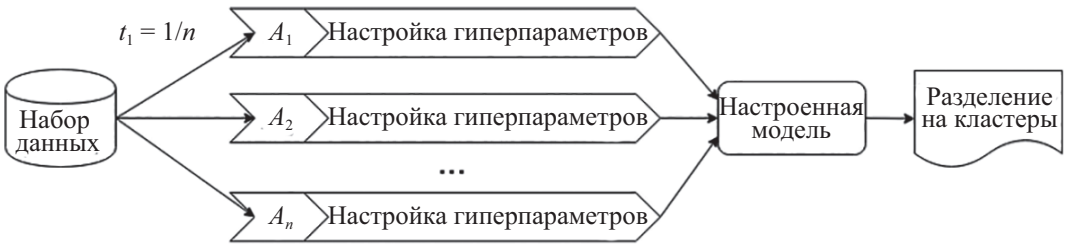

Рис. 1. Схема базового метода настройки гиперпараметров

Пусть для оптимизации гиперпараметров всех моделей доступен временной бюджет в T секунд. Заметим, что на разных машинах относительное количество времени, потраченного на вычисления, будет сохраняться. Исследуем вопрос распределения доступного временного бюджета между моделями.

Нельзя заранее предсказать, насколько качественную кластеризацию построит модель на конкретных данных. При разделении временного бюджета поровну между всеми моделями, т. е. полном приоритете на *исследование* их поведения, в итоге большая часть времени окажется в итоге потраченной на неэффективные модели. Другая крайность разделения времени — полный приоритет на *эксплуатацию* — состоит в выделении всего временного бюджета только одной модели, оказавшейся лучшей в самом начале. Тогда мы не рассматриваем алгоритмы других моделей, которые в итоге могут достигать лучшего качества кластеризации. Следовательно, необходимо найти компромисс между исследованием и эксплуатацией.

Поиск подобного компромисса является задачей обучения с подкреплением. Для его осуществления исходная задача сводится к задаче о многоруком бандите [23]. В задаче о многоруком бандите рассматривается агент с N ручками, с каждой из которых связано некоторое неизвестное распределение. В течение хода агент выбирает ручку и получает случайную награду из связанного с ней распределения. Цель агента — максимизировать полученную за k итераций награду.

Пусть задан некоторый временной бюджет T на поиск лучшего алгоритма $A^*_{\lambda^*}$. Требуется разбить его на интервалы $T = t_1 + t_2 + ... t_m$ таким образом, при запуске процессов π_j с ограничением по времени t_i мы получили значение функции качества Q такое, что: $Q(A^j_{\lambda_j}, D) \xrightarrow{(t_1, t_2, ... t_m)} \min_j$, где $A_j \in \mathbf{A}$, $\lambda = \pi(t_i, A^j, \varnothing)$ и $t_1 + ... + t_m = T$, $t_i \geq 0 \forall i$. В этом случае каждой ручке алгоритма многорукого бандита соответствует определенная модель алгоритма кластеризации из конечного множества $\mathbf{A}$, а вызову ручки i на итерации k — процесс оптимизации гиперпараметров этой модели в течение времени t_k. В результате будет достигнуто качество кластеризации $Q(A^j_{\lambda_j}, D)$, определяемое с помощью меры, оно и задает награду, получаемую по завершении итерации. Схема метода представлена на рис. 2.

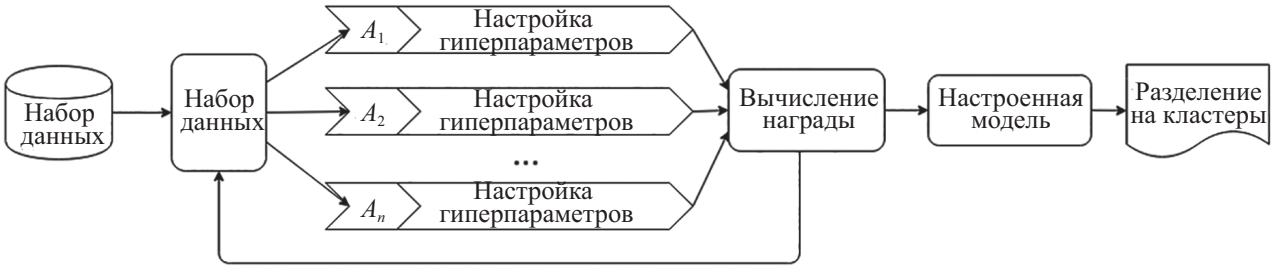

Рис. 2. Схема предложенного метода настройки гиперпараметров

Данный подход теоретически обоснован, так как предлагает решение задачи поиска компромисса между исследованием и эксплуатацией, который требуется найти в задаче одновременного выбора модели алгоритма кластеризации и ее гиперпараметров.

Здесь мы не рассмотрели выбор следующей ручки для запуска. Изначально этот шаг определялся с помощью известных алгоритмов решения задачи о многоруком бандите — Softmax и UCB1 [23]. Кроме того, значение суммарной награды ручки было заменено на значение средней награды, что при экспериментах обозначим RL-smx-tf, такое определение награды позволило убрать прямую зависимость от номера итерации.

Были предложены и другие методы определения следующей ручки на основе внутреннего устройства алгоритма SMAC, а именно *ожидаемого улучшения* (expected improvement, EI) и свойств случайного леса. В ходе работы SMAC оценивает конфигурации с помощью *ожидаемого улучшения*, значения которого предсказывает случайный лес.

В результате экспериментов было предложено еще несколько подходов, опишем только один из них, вошедший в итоговое сравнение. Назовем данный метод обучения с подкреплением *Softmax по нормализованным мерам и времени*. Решение о том, какую из ручек выбирает алгоритм обучения с подкреплением, принимается с помощью алгоритма Softmax, которому на итерации k подается на вход следующий вещественный вектор $\mathbf{X}$:

$$\mathbf{X} = S(\mathbf{R}^k) + S(\mathbf{U}^k),$$

где $\mathbf{R}^k = (r_i^k)_{i=1..n}$, $\mathbf{U}^k = \left(\sqrt{\frac{2\ln(n + \sum_{j=1}^{n} \ln t_j^k)}{1 + \ln t_j^k}} \right)_{i=1..n}$ и $S(x) = \left(\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_i}} \right)_{i=1..n}$.

Т. е. $\mathbf{R}$ — вектор наград, получаемых каждой ручкой обучения с подкреплением i; $\mathbf{U}$ — вектор поправок, которые вносятся в соответствии с затраченным временем для каждой ручки ; S — функция нормализации, аналогичная той, что используется внутри алгоритма Softmax.

Описание экспериментов

В экспериментах участвовало семь моделей алгоритмов: *k*-Means, DBSCAN, Affinity Propagation, Agglomerative Clustering, Mean Shift, Gaussian Mixture, Bayesian Gaussian Mixture. Эксперименты проводились на наборах реальных данных, находящихся в репозитории UCI и доступных по ссылке[1]. Для экспериментов использовалось шесть мер оценки разбиения: Calinski-Harabaz (CH), Sillhouette (Sil), sym, gd41, os, cop. CH и Sil были выбраны на основании проведенного исследования времени вычисления как требующие самых незначительных временных ресурсов [24]. Другие были выбраны согласно статье [25], так как они лучше всего соответствуют когнитивными представлениями людей-асессоров. В исследовании использован алгоритм оптимизации гиперпараметров SMAC, который минимизирует значение целевой функции, поэтому эксперименты нацелены на минимизацию меры качества. Значения некоторых мер увеличиваются с ростом качества, в этом случае они брались с противоположным знаком, что в данном случае равносильно. Следовательно, во всех последующих таблицах лучшим является меньшее значение.

Эксперименты запускались через систему slurlm[2], предназначенную для проведения экспериментов и управления ресурсами, из-за чего случайным образом распределялись планировщиком между серверами с 64-ядерными процессорами AMD Opteron 6378 @ 2.4 GHz, 256 GB RAM и AMD Opteron 6380 @ 2.5 GHz, 496 GB RAM. Все вычисления производились в один поток, на который выделялось до 2 ГБ оперативной памяти.

[1] http://archive.ics.uci.edu/ml/index.php
[2] https://slurm.schedmd.com

Экспериментальное сравнение базового метода и случайного поиска с вариациями метода обучения с подкреплением

В итоговое сравнение для наглядности было включено сравнение с качеством, достигаемым алгоритмами с параметрами *по умолчанию*. Кроме того, в него были включены случайный поиск (Random Search, RS), последовательный метод (Exhaustive Search, EX) и четыре предложенных вариации обучения с подкреплением.

Изначальный вариант обучения с подкреплением обозначим RL (reinforcement learning) — предложенный подход на основе решения задачи о многоруком бандите с алгоритмом Softmax по нормализованным мерам и времени.

На рис. 3 отображено сравнение работы этих трех подходов при различных временных бюджетах. Размер временного бюджета зависит от размера набора данных, так как при малом временном бюджете методы не успевают достичь осмысленных результатов на больших наборах данных.

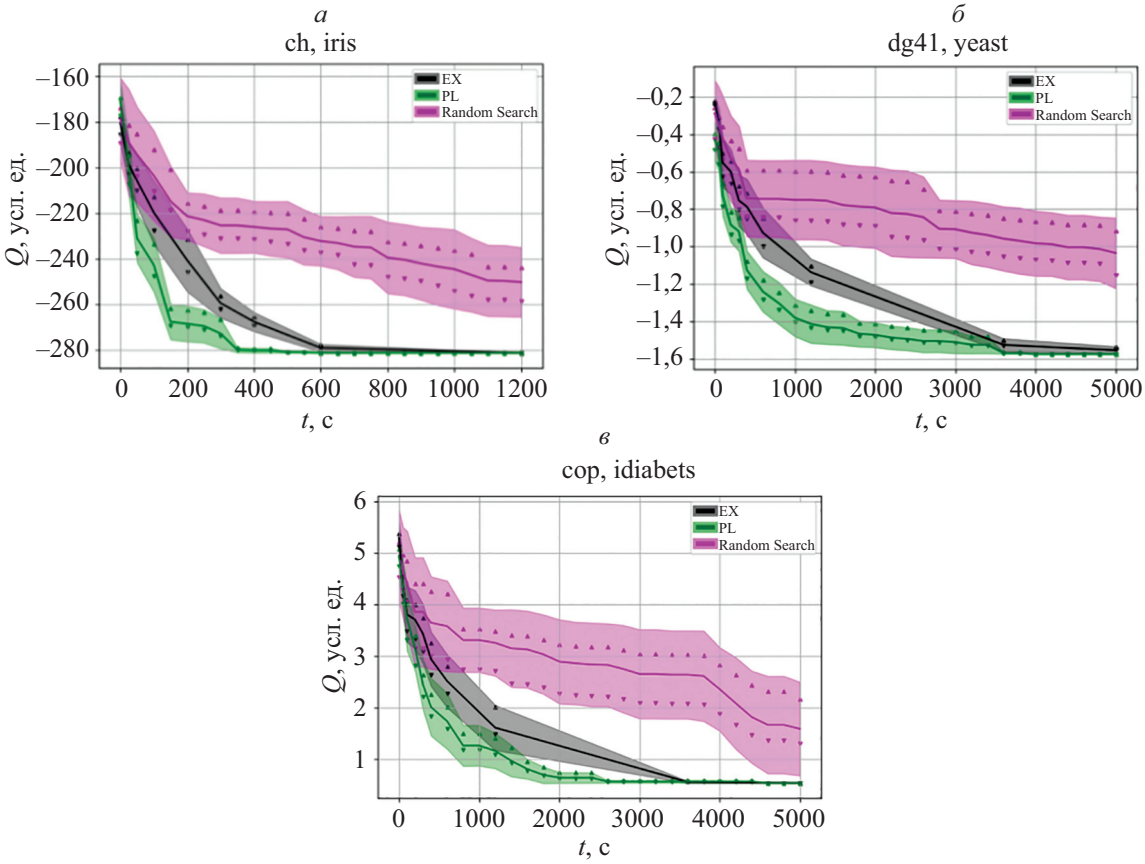

Рис. 3. Графики зависимостей значения выбранной меры Q от затраченного времени t. Линии на графике обозначают среднее значение целевой функции для каждого метода, затененная соответствующим цветом область — доверительный интервал, а треугольные символы — минимум и максимум из значений при данном временном бюджете: *а* — набор данных iris, мера Calinski-Harabasz; *б* — набор данных yeast, мера gd41; *в* — набор данных idiabets, мера cop

В таблице представлены численные значения мер при значении временного бюджета. Заметим, что запуск с большим временным бюджетом имеет смысл только для наборов данных большого размера, так как на малых наборах данных SMAC достигает лучшего результата быстрее, а потом не происходит никаких изменений.

Итоговый метод RL оказался статистически лучше базового на пяти мерах качества кластеризации из исследованных шести. На мере os, требующей наибольших затрат по времени, результаты получаются более неоднозначными и случайными, чем на остальных, т. е. нельзя с уверенностью сказать, какой из методов оказался лучше. Эксперименты показывают крайнюю вариативность значений меры os.

Из графиков и таблицы видно, что случайный поиск ожидаемо показал наихудшие результаты, а предложенный нами метод в среднем оказался лучше. Он раньше достигает минимума, а также в течение всего временного промежутка показывает лучшие результаты. Из этого можно заключить, что и при меньших временных бюджетах он превзойдет базовый метод полного перебора. Учет времени, затрачиваемого на запуск процессов-ручек, и нормализация награды помогают улучшить результаты.

Таблица. Численные значения мер при значении временного бюджета $T = 1200$.
RL-smx-tf — предложенный подход на основе решения задачи о многоруком бандите с алгоритмом Softmax.
RL-max-EI — предложенный подход на основе решения задачи о многоруком бандите с алгоритмом UCB1.
Жирным шрифтом отмечен минимальный результат

Мера	Набор данных	По умолчанию	RS	EX	RL	RL-max-EI	RL-smx-tf
ch	iris	–278,8425	–250,3009	**–281,3849**	**–281,3849**	**–281,3849**	**–281,3849**
ch	glass	–101,4215	–128,2044	**–177,7988**	**–177,7988**	–176,5833	**–177,7988**
ch	wholesale	–720,3281	–841,9137	–937,1773	**–937,1773**	–930,7756	**–937,1773**
ch	idiabets	–196,5603	–129,1122	–222,7155	**–236,3504**	–194,1084	–223,7569
ch	yeast	–264,5704	–220,0934	–296,1348	**–313,8025**	–275,7446	–295,4888
ch	krvskp	–266,6803	–167,3435	–239,7293	**–257,6422**	–192,4901	–243,5241
sil	iris	–0,60186	–0,512	**–0,6019**	**–0,6019**	–0,6002	**–0,6019**
sil	glass	–0,4987	–0,5042	**–0,5515**	**–0,5515**	–0,5471	**–0,5515**
sil	wholesale	–0,69	–0,6432	**–0,6897**	**–0,6897**	–0,6828	**–0,6897**
sil	idiabets	–0,2759	–0,3226	–0,485	**–0,5082**	–0,42	–0,4973
sil	yeast	–0,6032	–0,4961	–0,579	**–0,6531**	–0,4919	–0,6019
sil	krvskp	–0,1378	–0,0935	–0,1774	**–0,205**	–0,1077	–0,1794
sym	iris	–0,0018	–0,0022	**–0,0026**	**–0,0026**	–0,0025	**–0,0026**
sym	glass	–0,0014	–0,0012	**–0,0016**	**–0,0016**	**–0,0016**	**–0,0016**
sym	wholesale	–0,0007	–0,0014	**–0,0023**	**–0,0023**	–0,002	–0,0021
sym	idiabets	–0,0004	–0,0029	**–0,0045**	**–0,0045**	–0,0031	**–0,0045**
sym	yeast	–0,0003	–0,0002	–0,0003	**–0,0004**	–0,0003	–0,0003
sym	krvskp	–0,0002	–0,0001	–0,0001	**–0,0002**	–0,0001	–0,0001
gd41	iris	–0,9115	–1,033	**–1,1794**	**–1,1794**	–1,171	**–1,1794**
gd41	glass	–1,0928	–1,2415	–1,3835	**-1,3891**	–1,3625	**–1,3891**
gd41	wholesale	–1,2480	–0,6553	–1,1417	**–1,248**	–0,9297	–1,1442
gd41	idiabets	–1,4944	–1,0303	–1,5209	**–1,6061**	–1,2516	–1,5074
gd41	yeast	–1,4082	–0,7502	–1,1374	**–1,414**	–1,0358	–1,1901
gd41	krvskp	–1,1774	–0,6604	–0,6958	**–0,7591**	–0,4739	–0,7528
os	iris	–32,3439	–2156,962	–3117,5877	–3106,3131	**–3121,1775**	**–3121,1775**
os	glass	–491,5247	–665,8316	–870,8715	**–889,9995**	–862,5762	–864,0331
os	wholesale	–62,2361	–4084,9359	–6017,3197	**–6135,9152**	–4793,4681	–5253,2299
os	idiabets	–15642,84	–20453,783	–27841,9049	**–29445,211**	–24058,733	–24574,1892
os	yeast	–863,8433	–17632,971	**–21663,9475**	–18411,641	–19935,217	–20804,6337
os	krvskp	–300,551	–6949,9832	–6920,002	–5628,3257	**–7117,0635**	–5951,1009
cop	iris	0,605	2,3603	**0,605**	**0,605**	0,9535	**0,605**
cop	glass	2,2106	2,5235	0,6639	**0,6173**	1,8419	0,6173
cop	wholesale	1,9267	3,1852	1,4982	**0,8767**	2,3827	1,27
cop	idiabets	1,8873	3,2621	1,6182	**1,1682**	2,4276	1,4847
cop	yeast	2,3059	7,8689	5,5026	**3,3133**	6,3696	4,748
cop	krvskp	1,6455	2,3082	2,0637	**1,8845**	2,1271	1,8385

Заключение

В данной работе был предложен новый метод решения для задачи выбора модели алгоритмов кластеризации и настройки ее гиперпараметров. Необходимость настройки гиперпараметров алгоритмов показана экспериментально.

Было проведено масштабное исследование с целью выработать эффективный подход к нахождению компромисса в задаче выбора модели и настройки гиперпараметров: между исследованием большего числа моделей и более качественной эксплуатации каждой. Для поиска компромисса применено обучение с подкреплением и предложены разнообразные подходы, основанные как на целевых функциях и алгоритме

решения задачи о многоруком бандите, так и на внутреннем устройстве алгоритма SMAC. Было опробовано множество стратегий, и некоторые из них достигли статистически значимо лучших результатов по сравнению с базовыми методами, параметрами по умолчанию и случайным поиском.

Заметим, что рассмотренная эволюционная стратегия нахождения следующей конфигурации оказалась не лучше случайного и локального поиска, потому она была отвергнута. Изначально рассматривалась задача многокритериальной оптимизации, в решении которой эволюционные алгоритмы более эффективны. Но она не применима в реальности, так как обычно требуется оптимизация по одной-двум мерам качества.

Предложенный алгоритм был успешно протестирован на реальных задачах кластеризации и показал свою дееспособность. Он может применяться для любых задач кластеризации, также в нем применим любой метод оптимизации гиперпараметров.

В дальнейшем планируется улучшить предложенный метод с помощью добавления мета-обучения на стадии инициализации алгоритма решения задачи о многоруком бандите и для инициализации случайного леса.

Литература

1. Halkidi M., Batistakis Y., Vazirgiannis M. On clustering validation techniques // Journal of Intelligent Information Systems. 2001. V. 17. N 2–3. P. 107–145. doi: 10.1023/a:1012801612483
2. Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review // ACM Computing Surveys. 1999. V. 31. N 3. P. 264–323. doi: 10.1145/331499.331504
3. Mirkin B. Clustering for Data Mining: a Data Recovery Approach. CRC Press, 2005. 296 p. doi: 10.1201/9781420034912
4. Schlee D., Sneath P.H., Sokal R.R., Freman W.H. Numerical taxonomy. The principles and practice of numerical classification // Systematic Zoology. 1975. V. 24. N 2. P. 263–268. doi: 10.2307/2412767
5. Holzinger K.J., Harman H.H. Factor Analysis: A Synthesis of Factorial Methods. Chicago: University of Chicago Press, 1941. 417 p.
6. Chou C.H., Su M.C., Lai E. A new cluster validity measure and its application to image compression // Pattern Analysis and Applications. 2004. V. 7. N 2. P. 205–220. doi: 10.1007/s10044-004-0218-1
7. Luo M., Wang L.N., Zhang H.G. An unsupervised clustering-based intrusion detection method // Acta Electronica Sinica. 2003. V. 31. N 11. P. 1713–1716.
8. Von Luxburg U., Williamson R.C., Guyon I. Clustering: science or art // Proc. ICML Workshop on Unsupervised and Transfer Learning. Bellevue, USA, 2012. V. 27. P. 65–79.
9. Aggarwal C.C., Reddy C.K. Data Clustering: Algorithms and Applications. CRC press, 2013. 674 p. doi: 10.1201/b15410
10. Fraley C., Raftery A.E. How many clusters? Which clustering method? Answers via model-based cluster analysis // The Computer Journal. 1998. V. 41. N 8. P. 578–588. doi: 10.1093/comjnl/41.8.578
11. Hennig C. What are the true clusters? // Pattern Recognition Letters. 2015. V. 64. P. 53–62. doi: 10.1016/j.patrec.2015.04.009
12. Pal N.R., Biswas J. Cluster validation using graph theoretic concepts // Pattern Recognition. 1997. V. 30. N 6. P. 847–857. doi: 10.1016/s0031-3203(96)00127-6
13. Ferrari D.G., De Castro L.N. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods // Information Sciences. 2015. V. 301. P. 181–194. doi: 10.1016/j.ins.2014.12.044
14. Tibshirani R., Walther G., Hastie T. Estimating the number of clusters in a data set via the gap statistic // Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2001. V. 63. N 2. P. 411–423. doi: 10.1111/1467-9868.00293
15. Sugar C.A., James G.M. Finding the number of clusters in a dataset // Journal of the American Statistical Association. 2003. V. 98. N 463. P. 750–763. doi: 10.1198/016214503000000666
16. Pelleg D., Moore A.W. X-means: Extending k-means with efficient estimation of the number of clusters // Proc. 17th Int. Conf. on Machine Learning. Stanford, USA, 2000. Part 1. P. 727–734.
17. Thornton C., Hutter F., Hoos H.H., Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms // Proc. 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. Chicago, USA, 2012. P. 847–855. doi: 10.1145/2487575.2487629

References

1. Halkidi M., Batistakis Y., Vazirgiannis M. On clustering validation techniques. *Journal of Intelligent Information Systems*, 2001, vol. 17, no. 2–3, pp. 107–145. doi: 10.1023/a:1012801612483
2. Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review. *ACM Computing Surveys*, 1999, vol. 31, no. 3, pp. 264–323. doi: 10.1145/331499.331504
3. Mirkin B. *Clustering for Data Mining: a Data Recovery Approach*. CRC Press, 2005, 296 p. doi: 10.1201/9781420034912
4. Schlee D., Sneath P.H., Sokal R.R., Freman W.H. Numerical taxonomy. The principles and practice of numerical classification. *Systematic Zoology*, 1975, vol. 24, no. 2, pp. 263–268. doi: 10.2307/2412767
5. Holzinger K.J., Harman H.H. *Factor Analysis: A Synthesis of Factorial Methods*. Chicago, University of Chicago Press, 1941, 417 p.
6. Chou C.H., Su M.C., Lai E. A new cluster validity measure and its application to image compression. *Pattern Analysis and Applications*, 2004, vol. 7, no. 2, pp. 205–220. doi: 10.1007/s10044-004-0218-1
7. Luo M., Wang L.N., Zhang H.G. An unsupervised clustering-based intrusion detection method. *Acta Electronica Sinica*, 2003, vol. 31, no. 11, pp. 1713–1716.
8. Von Luxburg U., Williamson R.C., Guyon I. Clustering: science or art. Proc. *ICML Workshop on Unsupervised and Transfer Learning*. Bellevue, USA, 2012, vol. 27, pp. 65–79.
9. Aggarwal C.C., Reddy C.K. *Data Clustering: Algorithms and Applications*. CRC press, 2013, 674 p. doi: 10.1201/b15410
10. Fraley C., Raftery A.E. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 1998, vol. 41, no. 8, pp. 578–588. doi: 10.1093/comjnl/41.8.578
11. Hennig C. What are the true clusters? *Pattern Recognition Letters*, 2015, vol. 64, pp. 53–62. doi: 10.1016/j.patrec.2015.04.009
12. Pal N.R., Biswas J. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 1997, vol. 30, no. 6, pp. 847–857. doi: 10.1016/s0031-3203(96)00127-6
13. Ferrari D.G., De Castro L.N. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 2015, vol. 301, pp. 181–194. doi: 10.1016/j.ins.2014.12.044
14. Tibshirani R., Walther G., Hastie T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2001, vol. 63, no. 2, pp. 411–423. doi: 10.1111/1467-9868.00293
15. Sugar C.A., James G.M. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 2003, vol. 98, no. 463, pp. 750–763. doi: 10.1198/016214503000000666
16. Pelleg D., Moore A.W. X-means: Extending k-means with efficient estimation of the number of clusters. *Proc. 17th Int. Conf. on Machine Learning*. Stanford, USA, 2000, part 1, pp. 727–734.
17. Thornton C., Hutter F., Hoos H.H., Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proc. 19th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*. Chicago, USA, 2012, pp. 847–855. doi: 10.1145/2487575.2487629

18. Olson R.S., Bartley N., Urbanowicz R.J., Moore J.H. Evaluation of a tree-based pipeline optimization tool for automating data science // Proc. Genetic and Evolutionary Computation Conference. Denver, USA, 2016. P. 485–492. doi: 10.1145/2908812.2908918
19. Feurer M., Klein A., Eggensperger K., Springenberg J., Blum M., Hutter F. Efficient and robust automated machine learning // Advances in Neural Information Processing Systems. 2015. V. 2. P. 2962–2970.
20. Ефимова В.А., Фильченков А.А., Шалыто А.А. Применение обучения с подкреплением для одновременного выбора модели алгоритма классификации и ее структурных параметров // Машинное обучение и анализ данных. 2016. № 2. С. 244–254.
21. Kotthoff L., Thornton C., Hoos H.H., Hutter F., Leyton-Brown K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA // The Journal of Machine Learning Research. 2017. V. 18. N 1. P. 826–830.
22. Sutton R.S., Barto A.G. Introduction to Reinforcement Learning. Cambridge: MIT press, 1998. 322 p.
23. Hutter F., Hoos H.H., Leyton-Brown K. Sequential model-based optimization for general algorithm configuration // Lecture Notes in Computer Science. 2011. V. 6683. P. 507–523. doi: 10.1007/978-3-642-25566-3_40
24. Arbelaitz O., Gurrutxaga I., Muguerza J., Perez J. M., Perona I. An extensive comparative study of cluster validity indices // Pattern Recognition. 2003. V. 46. N 1. P. 243–256. doi: 10.1016/j.patcog.2012.07.021
25. Filchenkov A., Muravyov S., Parfenov V. Towards cluster validity index evaluation and selection // Proc. IEEE Artificial Intelligence and Natural Language Conference. St. Petersburg, Russia, 2016. P. 1–8.

18. Olson R.S., Bartley N., Urbanowicz R.J., Moore J.H. Evaluation of a tree-based pipeline optimization tool for automating data science. *Proc. Genetic and Evolutionary Computation Conference*. Denver, USA, 2016, pp. 485–492. doi: 10.1145/2908812.2908918
19. Feurer M., Klein A., Eggensperger K., Springenberg J., Blum M., Hutter F. Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems*, 2015, vol. 2, pp. 2962–2970.
20. Efimova V.A., Filchenkov A.A., Shalyto A.A. Reinforcement-based simultaneous classification model and its hyperparameters selection. *Machine Learning and Data Analysis*, 2016, no. 2, pp. 244–254. (in Russian)
21. Kotthoff L., Thornton C., Hoos H.H., Hutter F., Leyton-Brown K. Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *The Journal of Machine Learning Research*, 2017, vol. 18, no. 1, pp. 826–830.
22. Sutton R.S., Barto A.G. *Introduction to Reinforcement Learning.* Cambridge, MIT press, 1998, 322 p.
23. Hutter F., Hoos H.H., Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. *Lecture Notes in Computer Science*, 2011, vol. 6683, pp. 507–523. doi: 10.1007/978-3-642-25566-3_40
24. Arbelaitz O., Gurrutxaga I., Muguerza J., Perez J. M., Perona I. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 2003, vol. 46, no. 1, pp. 243–256. doi: 10.1016/j.patcog.2012.07.021
25. Filchenkov A., Muravyov S., Parfenov V. Towards cluster validity index evaluation and selection. *Proc. IEEE Artificial Intelligence and Natural Language Conference*. St. Petersburg, Russia, 2016, pp. 1–8.

Авторы

Муравьёв Сергей Борисович — программист, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57194035005, ORCID ID: 0000-0002-4251-1744, mursmail@gmail.com

Ефимова Валерия Александровна — программист, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, ORCID ID: 0000-0002-5309-2207, valeryefimova@gmail.com

Шаламов Вячеслав Владимирович — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 57191077141, ORCID ID: 0000-0002-5647-6521, sslavian812@gmail.com

Фильченков Андрей Александрович — кандидат физико-математических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 55507568200, ORCID ID: 0000-0002-1133-8432, aaafil@mail.ru

Сметанников Иван Борисович — кандидат технических наук, ассистент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 56902520900, ORCID ID: 0000-0003-3376-9468, smeivan@mail.ru

Authors

Sergey B. Muravyov — Software engineer, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57194035005, ORCID ID: 0000-0002-4251-1744, mursmail@gmail.com

Valeria A. Efimova — Software engineer, ITMO University, Saint Petersburg, 197101, Russian Federation, ORCID ID: 0000-0002-5309-2207, valeryefimova@gmail.com

Vyacheslav V. Shalamov — postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 57191077141, ORCID ID: 0000-0002-5647-6521, sslavian812@gmail.com

Andrey A. Filchenkov — PhD, Associate Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 55507568200, ORCID ID: 0000-0002-1133-8432, aaafil@mail.ru

Ivan B. Smetannikov — PhD, Assistant, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 56902520900, ORCID ID: 0000-0003-3376-9468, smeivan@mail.ru