УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
сентябрь–октябрь 2020 Том 20 № 5 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
September–October 2020 Vol. 20 No 5 ISSN 2226-1494 http://ntv.ifmo.ru/en/
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

УДК 004.852 doi: 10.17586/2226-1494-2020-20-5-667-676

МЕТОД ВЫБОРА ГИПЕРПАРАМЕТРОВ В ЗАДАЧАХ МАШИННОГО ОБУЧЕНИЯ ДЛЯ КЛАССИФИКАЦИИ СТОХАСТИЧЕСКИХ ОБЪЕКТОВ

А.В. Тимофеев

ТОО «Эквалайзум», Астана, 010000, Казахстан
Адрес для переписки: timofeev.andrey@gmail.com

Информация о статье
Поступила в редакцию 01.07.20, принята к печати 10.08.20
Язык статьи — русский

Ссылка для цитирования: Тимофеев А.В. Метод выбора гиперпараметров в задачах машинного обучения для классификации стохастических объектов // Научно-технический вестник информационных технологий, механики и оптики. 2020. Т. 20. № 5. С. 667–676. doi: 10.17586/2226-1494-2020-20-5-667-676

Аннотация

Предмет исследования. Предложен простой и эффективный метод выбора гиперпараметров при решении классификационной проблемы методами машинного обучения. Метод работает с любыми гиперпараметрами вещественного типа, значения которых лежат внутри известного вещественного параметрического компакта. **Метод.** Внутри параметрического компакта генерируется случайная выборка (пробная сеть) сравнительно небольшого объема, для каждого элемента которой вычисляется эффективность выбора гиперпараметров согласно специальному критерию. Эффективность оценивается величиной некоторого вещественного скаляра, который принципиально не зависит от порога классификации. Таким образом, формируется выборка регрессии, регрессорами которой являются случайные наборы гиперпараметров из параметрического компакта, а значениями регрессии – соответствующие этим наборам значения показателя эффективности классификации. На основании полученной выборки строится непараметрическая аппроксимация этой регрессии. На следующем этапе, используя метод оптимизации Нелдера–Мида, определяется минимальное значение построенной аппроксимации для регрессионной функции на параметрическом компакте. Аргументы минимального значения регрессии являются приближенным решением поставленной задачи. **Основные результаты.** В отличие от традиционных, предложенный подход основан на непараметрической аппроксимации функции регрессии: набор гиперпараметров — значение показателя эффективности классификации. Особое внимание уделено выбору критерия качества классификации. За счет использования аппроксимации упомянутого типа имеется возможность исследования поведения показателя эффективности вне значений пробной сетки. Как следует из проведенных экспериментов на различных базах данных, предложенный подход обеспечивает существенный прирост эффективности выбора гиперпараметров по сравнению с базовыми вариантами и одновременно сохраняет практически приемлемую работоспособность даже для малых значений мощности пробного множества. Новизна подхода заключена в одновременном использовании: непараметрической аппроксимации для функции регрессии, которая связывает значения гиперпараметров с соответствующими им величинами критерия качества; выборе критерия качества классификации и метода поиска глобального экстремума этой функции. **Практическая значимость.** Предложенный алгоритм выбора гиперпараметров может быть использован в любых системах, основанных на принципе машинного обучения. Например, в системах управления технологическими процессами, биометрических системах и системах машинного зрения.

Ключевые слова
выбор гиперпараметров, машинное обучение, Multiclass Gradient Boosting Classifier, Multiclass SVM-classifier, SV-регрессия, Gradient Boosting Regression, метод Нелдера–Мида

doi: 10.17586/2226-1494-2020-20-5-667-676

METHOD FOR HYPERPARAMETER TUNING IN MACHINE LEARNING TASKS FOR STOCHASTIC OBJECTS CLASSIFICATION

A.V. Timofeev

LLP EqualiZoom, Astana, 010000, Republic of Kazakhstan
Corresponding author: timofeev.andrey@gmail.com

Article info
Received 01.07.20, accepted 10.08.20
Article in Russian

For citation: Timofeev A.V. Method for hyperparameter tuning in machine learning tasks for stochastic objects classification. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2020, vol. 20, no. 5, pp. 667–676 (in Russian). doi: 10.17586/2226-1494-2020-20-5-667-676

Abstract

Subject of Research. The paper presents a simple and practically effective solution for hyperparameter tuning in classification problem by machine learning methods. The proposed method is applicable for any hyperparameters of the real type with the values which lie within the known real parametric compact. **Method.** A random sample (trial network) of small power is generated within the parametric compact, and the efficiency of hyperparameter tuning is calculated for each element according to a special criterion. The efficiency is estimated by the value of a real scalar, which does not depend on the classification threshold. Thus, a regression sample is formed, the regressors of which are the random sets of hyperparameters from the parametric compact, and regression values are classification efficiency indicator values corresponding to these sets. The nonparametric approximation of this regression is constructed on the basis of the formed data set. At the next stage the minimum value of the constructed approximation is determined for the regression function on the parametric compact by the Nelder-Mead optimization method. The arguments of the minimum regression value appear to be an approximate solution to the problem. **Main Results.** Unlike traditional approaches, the proposed approach is based on non-parametric approximation of the regression function: a set of hyperparameters – classification efficiency index value. Particular attention is paid to the choice of the classification quality criterion. Due to the use of the mentioned type approximation, it is possible to study the performance indicator behavior out of the trial grid values ("between" its nodes). As it follows from the experiments carried out on various databases, the proposed approach provides a significant increase in the efficiency of hyperparameter tuning in comparison with the basic variants and at the same time maintains almost acceptable performance even for small values of the trial grid power. The novelty of the approach lies in the simultaneous use of non-parametric approximation for the regression function, which links the hyperparameter values with the corresponding values of the quality criterion, selection of the classification quality criterion, and search method for the global extremum of this function. **Practical Relevance.** The proposed algorithm for hyperparameters tuning can be used in any systems built on the principles of machine learning, for example, in process control systems, biometric systems and machine vision systems.

Keywords

hyperparameters tuning, machine learning, multiclass gradient boosting classifier, multiclass SVM-classifier, SV-regression, gradient boosting regression, Nelder-Mead method

Введение

Выбор значений так называемых гиперпараметров мультиклассовых классификаторов представляет собой принципиально важную проблему, так как гиперпараметры в значительной мере формируют структуру классификатора, и поэтому определяют границы его практической эффективности. Исторически используются различные методы определения гиперпараметров мультиклассовых классификаторов, в том числе: поиск по регулярной решетке [1]; случайный поиск [2]; байесовская оптимизация [3–5] (с использованием «суррогатной модели»), оптимизация на основе градиентов [6] и некоторые другие. Предлагаемый в настоящей работе метод представляет собой один из вариантов байесовской оптимизации, в котором, в отличие от известных подходов, в качестве суррогатной функции выступает непараметрическая регрессия, построенная на базе множества предварительно вычисленных значений обобщенного критерия качества классификатора. Собственно регрессорами являются значения гиперпараметров в предположении, что они принадлежат некоторому, априорно заданному параметрическому компакту. Используя оптимизационную процедуру, которая не подразумевает необходимости вычисления градиентов оптимизируемого функционала, определяется минимальное значение нелинейной регрессии на параметрическом компакте. Аргументы минимума регрессии являются приближенным решением поставленной задачи. Для того чтобы обеспечить глобальность решения оптимизационной задачи, алгоритм оптимизации запускается несколько раз, с различными начальными условиями. Тестирование метода на нескольких тестовых наборах данных показало высокую эффективность предложенного подхода.

Основные определения и понятия

Наблюдаются процесс $\{x_i^{(\theta^*)} | l \geq 0\}$, имплицитно зависящий от априорно неизвестного параметра (индекса) θ^*, $\theta^* \in \Theta$, где $\Theta = \{\theta_k\} \subseteq R^1$, $|\Theta| < \infty$ такое, что $\theta_2 \neq \theta_1 \in \Theta$, — априорно заданное множество индексов. $x_l^{(\theta)} \in X$, где X — множество значений наблюдаемого процесса, а (X, d_x) — компактное метрическое пространство наблюдений; d_x — метрика этого пространства. Обозначим (Z, d_z) — компактное метрическое пространство признаков, где d_z — метрика этого пространства; Z — множество значений признаков. Элементами этого пространства являются наборы числовых параметров, которые информативно характеризуют наблюдения. При решении классификационной задачи пространство (X, d_x) отображается в пространство (Z, d_z) при помощи некоторого известного преобразования L, т. е. $\forall x \in X \; \exists \, z \in Z : z = L(x)$. Так как основные манипуляции, которые интересуют автора в рамках настоящей работы, совершаются в пространстве (Z, d_z), для удобства будем считать, что наблюдения сразу проецируются в пространство признаков. Для каждого класса из Θ заданы обучающие корпуса:

$$Z_{tr}^{(\theta)} = \{(z_{tr,j}^{(\theta)}, \theta) | j = 1, \ldots, m_\theta\} \subseteq \mathbf{Z}, \theta \in \Theta,$$
$$\forall \theta \in \Theta: |Z_{tr}^{(\theta)}| = m_\theta < \infty, z_{tr} = \bigcup_{\theta \in \Theta} Z_{tr}^{(\theta)}.$$

Здесь $z_{tr,l}^{(\theta)} = F(x_{tr,l}^{(\theta)})$, $x_{tr,l}^{(\theta)} \in X$, $z_{tr,l}^{(\theta)} \in Z$. Классификатор $f(\beta|h)$: $Z \to \Theta$ — это некая функция, которая разделяет пространство (Z, d) на $m = |\Theta| = \sum_{\theta \in \Theta} m_\theta$ классов. Классификатор зависит от параметров двух типов: $\beta \in B$ — настраиваемые параметры и $h \in H$ — так называемые гиперпараметры, которые в значительной мере определяют структуру классификатора, где B и

H — известные компакты. Основное отличие параметров β и h, с точки зрения теории машинного обучения, заключается в том, что настраиваемые параметры β могут быть определены только при фиксированной структуре классификатора. В связи с этим сначала путем выбора конкретного значения набор гиперпараметров $h \in H$ фиксируется структура классификатора $f(\beta|h)$, и только после этого производится подстройка параметров β путем обучения на базе $\mathbf{Z}_{tr}$. Для определения оптимальной пары параметров (β, h) требуется повторить данную процедуру многократно.

Пусть фиксирован набор гиперпараметров $h \in H$. В этом случае параметры $\beta \in \mathrm{B}$ классификатора $f(\beta|h)$ выбираются при помощи поисковой процедуры $S_f(\beta|h, \mathbf{Z}_{tr})$. Эта процедура зависит от типа классификатора $f(\cdot)$ и, как правило, сводится к определению такого набора $\beta^* \in \mathrm{B}$, который доставляет минимум ошибок классификации:

$$Q(\mathbf{Z}_T, f(\beta|h)) = \left\langle \left\langle [f(\beta|h)(z_{tr}^{(\theta)}) \neq \theta] \right\rangle_{z_{tr}^{(\theta)} \in Z_T} \right\rangle_{\theta \in \Theta}, \; Z_T \subseteq \mathbf{Z}_{tr},$$

при ограничениях, в качестве которых выступают условия регуляризации, где Z_T — множество прецедентов, на котором классификатор $f(\cdot)$ был обучен; $\langle\rangle$ — функция усреднения; $[\cdot]$ — скобка Айверсона. Смысл регуляризации непрост и сводится к управлению сложностью модели классификатора так, чтобы функция $f(\beta|h)$: $Z \to \Theta$ не была слишком «сложной» в смысле теории Вапника–Червоненкиса. Управление сложностью классификатора призвано повысить его обобщающую способность, ограничивая степень его переобученности на конкретном обучающем наборе $\mathbf{Z}_{tr}$. Кроме того, для повышения обобщающей способности зачастую используется ресемплинг на наборе $\mathbf{Z}_{tr}$, в том числе популярный метод перекрестной проверки (Cross Validation). Регуляризация может быть в явном виде отражена в функционале ошибок, например, для классификатора типа Support Vector Machine регуляризирующий компонент присутствует в виде аддитивного члена, что хорошо согласуется с принципом регуляризации по Тихонову. В других случаях регуляризация может выражаться в качестве одного из компонентов гиперпараметра $h \in H$, например: в качестве параметра «максимальное количество листьев» для классификатора типа «случайный лес» (Random Forest Classifier).

Таким образом, тип классификатора существенным образом определяет структуру процедуры $S_f(\beta|h, \mathbf{Z}_{tr})$. Например, в случае классификатора типа «нейронная сеть» — эта процедура называется «метод обратного распространения ошибки». Данная процедура постоянно модифицируется, реагируя на появление новых топологий нейронных сетей, и в настоящее время представляет собой достаточно эффективный алгоритм.

Как уже отмечалось ранее, набор гиперпараметров $h \in H$ кардинальным образом характеризует структуру классификатора. В этом состоит основное отличие гиперпараметров от настраиваемых параметров $\beta \in \mathrm{B}$, значения которых определяют качество работы классификатора при фиксированном наборе $h \in H$ для заданного корпуса обучающих данных $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$ (здесь $\mathbf{Z}$ — все возможные корпуса типа $\mathbf{Z}_{tr}$). Например, в случае нейронной сети гиперпараметрами являются: количество слоев, число нейронов в каждом слое, тип функции активации и прочие параметры, которые определяют собственно топологию сети. Заметим, что многие известные классификаторы, которые изначально исследовались в теории обособленно (например, Support Vector Machine, Extreme Learning Machine, сеть Кохонена и др.) или допускают эквивалентную интерпретацию в виде нейронной сети, или попросту являются нейронными сетями с конкретной топологией. Эта особенность подчеркивает важность проблемы выбора гиперпараметров, в особенности для случая нейронных сетей, так как в данном случае гиперпараметры однозначно определяют топологию конкретной нейронной сети, которая обуславливает ее потенциальные возможности и ограничения.

Обозначим $\mathbf{\Omega} = B \otimes H$ – область допустимых значений параметров $\beta \in \mathrm{B}$, $h \in H$. С точки зрения процесса аппроксимации разделяющих классы гиперповерхностей $\{D_\theta|\theta \in \Theta\}$, к которой сводится процедура обучения любого классификатора, допустима условная аналогия с понятиями локального и глобального экстремумов выбранного критерия качества аппроксимации этих «идеальных» гиперплоскостей. Аппроксимация гиперповерхностей $\{D_\theta|\theta \in \Theta\}$ производится путем подгонки параметров $\beta \in \mathrm{B}$ классификатора $f(\beta|h)$ на обучающем корпусе $\mathbf{Z}_{tr}$ при помощи специальной поисковой процедуры $S_f(h|\beta, \mathbf{Z}_{tr})$, для фиксированного $h \in H$. В результате этой аппроксимации, явно или неявно, формируются аппроксимирующие гиперплоскости $\{d_{\theta,f}(\beta|h, \mathbf{Z}_{tr})|\theta \in \Theta\}$. Существуют ситуации, когда в выбранном пространстве признаков некоторые классы из Θ – принципиально неразличимы. Для простоты эти ситуации будут исключены из рассмотрения, т. е. полагаем, что для любого класса из Θ может быть построена гиперповерхность D_θ, которая идеально отделяет этот класс от других. При этом степень качества аппроксимации $d_{\theta,f}(\beta|h, \mathbf{Z}_{tr})$ для D_θ оценивается при помощи некоторого функционала $I_f(D_\theta, d_{\theta,f}(\beta|h, \mathbf{Z}_{tr}))$ такого, что

$$\forall \theta \in \Theta, h \in H,$$
$$\mathbf{Z}_{tr} \subseteq \mathbf{Z}: [0 \le m_\theta(\mathbf{Z}_{tr}) \le I_f(D_\theta, d_{\theta,f}(\beta|h, \mathbf{Z}_{tr})) \le a \le 1],$$

где $m_\theta(\mathbf{Z}_{tr})$ — потенциально достижимая точность аппроксимации для корпуса $\mathbf{Z}_{tr}$ и класса $\theta \in \Theta$; a — некоторое число, $0 \le a \le 1$. Функционал $I_f(\cdot)$ иначе будем называть критерием качества аппроксимации границ. Чем лучше аппроксимация, тем меньше значения функционала $I_f(\cdot)$, идеальная аппроксимация (для корпуса $\mathbf{Z}_{tr}$) соответствует случаю $I_f(\cdot) = m_\theta(\mathbf{Z}_{tr})$. $(\mathbf{Z}_{tr} = \mathbf{Z}) \Rightarrow$ $\Rightarrow (I(\cdot) = 0)$. Качество аппроксимации на всем множестве Θ характеризуется функционалом:

$$I_f(\beta, \Theta, h, \; \mathbf{Z}_{tr}) = \langle I_f(D_\theta, d_{\theta,f}(\beta|h, \mathbf{Z}_{tr}))\rangle_{\theta \in \Theta}.$$

Обозначим: $S(\mathbf{\Omega}, B, \beta_0, Q(\mathbf{Z}_{tr}, f(\beta|h))|\Theta) \equiv S_f(h|B, \mathbf{Z}_{tr})$ — процедура поиска локального экстремума функционала ошибок $Q(\mathbf{Z}_{tr}, f(\beta|h))$, по аргументу β, для обучающего корпуса $\mathbf{Z}_{tr}$ и классификатора $f(\beta|h)$, при фиксированном h; β_0 — начальное значение аргументов поиска (точка старта) при фиксированном $h \in H$, $(\beta_0, h) \in \mathbf{\Omega}$. Как правило, процедура $S_f(\cdot)$ реализуется по схеме

жадной оптимизации (Greedy Algorithm). В результате работы процедуры $S_f(\cdot)$ получаем набор настроечных параметров $\beta^*(\mathbf{Z}_{tr}|h) \in B$ для $f(\beta|h)$, $(\beta^*(\mathbf{Z}_{tr}|h), h) \in \mathbf{\Omega}$. Пусть $\mathbf{\Omega}$ для любого корпуса $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$ содержит точку $(\beta^*(\mathbf{Z}_{tr}|h), h)$, для которой $(\beta_f^*(\mathbf{Z}_{tr}|h) = \operatorname{Arg}\inf_{\beta \in B} Q(\mathbf{Z}_{tr}, f(\beta|h))$. Так как процедура $S_f(\cdot)$ имеет локальный характер, ее результат зависит от начальных условий β_0. По этой причине в общем случае допустима запись:

$$\forall \mathbf{Z}_{tr} \subseteq \mathbf{Z}\ \exists\ B'(h)\colon (\beta_0 \in B'(h)) \Rightarrow$$
$$\Rightarrow (S_f(\mathbf{\Omega}, B, \beta_0, Q(\mathbf{Z}_{tr}, f(\beta|h))|\Theta) = \beta_f^*(\mathbf{Z}_{tr}|h)).$$

Таким образом, в результате работы процедуры $S_f(\cdot)$, для заданных $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$, $h \in H$, будут получены аппроксимации $d_{\theta,f}(\beta^*(\mathbf{Z}_{tr}|h)|h, \mathbf{Z}_{tr})$ гиперповерхностей D_θ, разделяющих классы $\theta \in \Theta$. Аппроксимации $d_{\theta,f}(\cdot|h, \mathbf{Z}_{tr})$ существенным образом зависят от корпуса $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$. В общем случае:

$$\forall h \in H, \theta \in \Theta, \mathbf{Z}_1 \neq \mathbf{Z}_2 \subseteq \mathbf{Z}\colon d_{\theta,f}(\beta^*(\mathbf{Z}_1|h)|h, \mathbf{Z}_1) \neq$$
$$\neq d_{\theta,f}(\beta^*(\mathbf{Z}_2|h)|h, \mathbf{Z}_2).$$

Собственно говоря, в этом и состоит проблема обеспечения высокой обобщающей способности процедуры обучения классификатора $f(\beta|h)$. Дело в том, что высокая точность аппроксимации гиперплоскостей $\{D_\theta|\theta \in \Theta\}$, полученная на основе использования конкретного корпуса $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$ в виде аппроксимаций $\{d_{\theta,f}(\beta^*(\mathbf{Z}_{tr}|h)|h, \mathbf{Z}_{tr})|\theta \in \Theta\}$, не гарантирует того, что на другом корпусе $\mathbf{Z}_{tr}^{\#} \neq \mathbf{Z}_{tr}$, $\mathbf{Z}_{tr}^{\#} \subseteq \mathbf{Z}$ данные аппроксимации будут эффективными. В этой связи для конкретного корпуса $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$ нет смысла строить оценку слишком высокого качества. Так как повышение качества аппроксимации на корпусе $\mathbf{Z}_{tr}$ с высокой вероятностью снижает качество аппроксимации на корпусах из множества $\mathbf{Z}\backslash\mathbf{Z}_{tr}$. По этой причине, в функционал ошибок $Q(\mathbf{Z}_{tr}, f(\beta|h))$ иногда добавляют специальные регуляризирующие параметры, которые искусственно понижают точность аппроксимации и которые, по сути дела, являются гиперпараметрами.

Обозначим:

$$\{\beta_f^{***}(h_f^{***}), h_f^{***}\} = \operatorname{Arg}\inf_{\beta, h \in \mathbf{\Omega}} I_f(D_\theta, d_{\theta,f}(\beta|h, \mathbf{Z})),$$

где $\{\beta_f^{***}(h_f^{***}), h_f^{***}\}$ — оптимальная параметрическая пара, которую можно определить для $f(\beta|h)$ только на всем множестве возможных корпусов $\mathbf{Z}$, что практически невозможно. На практике приходится обходиться оценками этой пары, которая определяется на базе доступных корпусов, учитывая необходимость сохранения высокой обобщающей способности классификатора. В том случае, когда задан корпус $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$, имеем:

$$c_f(\mathbf{Z}_{tr}) = \{\beta_f^{**}(h_f^{**}), h_f^{**}\} =$$
$$= \operatorname{Arg}\inf_{\beta, h \in \mathbf{\Omega}} I_f(D_\theta, d_{\theta,f}(\beta|h, \mathbf{Z}_{tr})), I_f(D_\theta, d_{\theta,f}(\beta^{**}(\mathbf{Z}_{tr}, h^{**})) \geq$$
$$\geq m_\theta(\mathbf{Z}_{tr}).$$

В данном случае $c_f(\mathbf{Z}_{tr})$ представляет собой параметрическую пару, которая:

1) является условно оптимальным решением аппроксимационной задачи для классификатора $f(\beta|h)$;
2) соответствует корпусу $\mathbf{Z}_{tr} \subseteq \mathbf{Z}$;
3) является потенциально достижимой целью при настройке параметров классификатора $f(\beta|h)$ в условиях, когда полный корпус данных $\mathbf{Z}$ — недоступен.

При удачно выбранном начальном условии β_0, имеем: $\beta_f^{**}(\mathbf{Z}_{tr}|h_f^{**}) = S_f(h_f^{**}|B, \mathbf{Z}_{tr})$.

Теперь, когда определены основные термины и понятия, можно перейти к основной задаче настоящей работы: определению практически эффективных оценок для пары $c_f(\mathbf{Z}_{tr})$. Так как величины $\beta_f^{**}(\mathbf{Z}_{tr}|h_f^{**})$ и h_f^{**} — взаимозависимы согласно способу их определения, на практике параметрическая пара $c_f(\mathbf{Z}_{tr})$ вычисляется итерационно, на базе использования некоторой эмпирической процедуры $O_f(\mathbf{Z}_{tr}, H)$, которая тем или иным образом основана на исследовании зависимости функционала качества аппроксимации $\mathbf{I}_f(\beta(h), \Theta, h, \mathbf{Z}_{tr})$ от значений гиперпараметра h на H.

Исследование такого рода невозможно без вычисления множества $\{\beta_f(\mathbf{Z}_{tr}|h)|h \in \mathbf{h}\} \subseteq B$, которое определяется с использованием процедуры $S_f(\cdot)$ на некотором множестве «проб» $\mathbf{h} = \{h|h \in H\}$, $|\mathbf{h}| < \infty$. Для конкретного $h \in H$, стоимость вычисления набора $\beta_f(\mathbf{Z}_{tr}|h) \equiv \beta_f(h)$ при помощи процедуры $S_f(h|B, \mathbf{Z}_{tr})$ может быть достаточно велика. Именно по этой причине величина $|\mathbf{h}|$ не может быть значительной и, как правило, $n = |\mathbf{h}| < 50$. Техническая проблема состоит в том, что функционал качества $\mathbf{I}_f(\beta_f(h), \Theta, h, \mathbf{Z}_{tr})$ не может быть вычислен в явном виде, так как множество $\{D_\theta|\theta \in \Theta\}$ всегда неизвестно. В связи с этим на практике требуется некоторая аппроксимация $\Upsilon_f(\beta_f(h), h|\Theta, \mathbf{Z}_{tr})$ этого функционала, которая, в идеале, характеризует качество работы алгоритма $f(\beta|h)$ на Θ, зависит от $h \in H$ и Θ, но не зависит от значения порога принятия решения. Выбору подходящей аппроксимации $\Upsilon_f(\beta_f(h), h|\Theta, \mathbf{Z}_{tr})$ посвящен следующий раздел статьи. В дальнейшем будем вместо $\Upsilon_f(\beta_f(h), h|\Theta, \mathbf{Z}_{tr})$ использовать сокращенную запись $\Upsilon_f(\beta_f(h), h)$.

Будем учитывать эти ограничения, обозначая эмпирическую процедуру $O_f(\mathbf{Z}_{tr}, H)$ для вычисления $c_f(\mathbf{Z}_{tr})$ при фиксированном n следующим образом: $O_{n,f}(\mathbf{Z}_{tr}, \mathbf{h})$. Естественно, что ввиду конечности $|\mathbf{h}|$ точное вычисление $c_f(\mathbf{Z}_{tr})$ принципиально невозможно, а на выходе вычислительной процедуры получено некоторое приближение, которое обозначим $c_{n,f}(\mathbf{Z}_{tr})$: $c_{n,f}(\mathbf{Z}_{tr}) =$ $= \{\beta_{n,f}(\mathbf{Z}_{tr}|h_{n,f}), h_{n,f}\} = O_{n,f}(\mathbf{Z}_{tr}, \mathbf{h})$.

Пусть заданы несколько Data Set, которые совместно составляют множество $\mathbf{DS} = \{DS_j|j = 1,3\}$ и представляют собой размеченные корпуса мультиклассовых наблюдений. Корпуса **DS** будут использованы для тестирования предлагаемого алгоритма. Структура данных по этим корпусам будет подробно описана в дальнейшем тексте статьи.

Разработчики каждого алгоритма классификации $f(\beta|h)$, как правило, всегда рекомендуют некоторые «номинальные» (базовые) значения его гиперпараметров, которые не зависят от обучающих корпусов. Обозначим базовое значение символом $h_{B,f}$ и будем использовать его в качестве **опорного решения** при сравнительном анализе. Кроме того, для сравнительного анализа будет использован и широко известный метод оптимизации гиперпараметров при помощи случайного поиска [2], который обозначим символом $h_{ST,f,n}^{(j)}$ (второе опорное

решение, которое зависит от соответствующего корпуса $DS_j \in \mathbf{DS}$ и числа проб n). В этом случае $\beta_{B,f}(h_{B,f}) = S_f(h_{B,f}|B, \mathbf{Z}_{tr})$, $\beta_{ST,f}(h^{(j)}_{ST,f,n}) = S_f(h^{(j)}_{ST,f,n}|B, \mathbf{Z}_{tr})$.

Постановка задачи

Для заданных $\mathbf{Z}_{tr}$, H и $f(\beta|h)$ необходимо определить такую процедуру $O_{n,f}(\mathbf{Z}_{tr}, \mathbf{h})$, которая позволяет, при минимальном $n = |\mathbf{h}|$, для $\mathbf{Z}_{j,tr} \subseteq DS_j$, $DS_j \in \mathbf{DS}$, вычислять оценку $c_{n,f,j}(\mathbf{Z}_{j,tr}) = \{\beta_{n,f,j}(\mathbf{Z}_{j,tr}|h^{(j)}_{n,f}), h^{(j)}_{n,f}\}$ так, что

$$\forall DS_j \in \mathbf{DS}: [\Upsilon_f(\beta_{n,f}(h^{(j)}_{n,f}), h^{(j)}_{n,f}) < \Upsilon_f(\beta_{B,f}(h_{B,f}), h_{B,f}),$$
$$\Upsilon_f(\beta_{n,f}(h^{(j)}_{n,f}), h^{(j)}_{n,f}) < \Upsilon_f(\beta_{ST,f}(h^{(j)}_{ST,f,n}), h^{(j)}_{ST,f,n})].$$

Критерий эффективности (информативности)

Крайне важен выбор критерия эффективности (информативности) $\Upsilon_f(\beta(h), h)$ классификационной процедуры $f(\beta|h)$. Существует множество взаимодополняющих друг друга критериев, в том числе, например: вероятность правильной классификации, accuracy, «сбалансированная точность» (balanced accuracy), TPR (true positive rate), F_β score ($\beta = 1,2,..$), AUC-ROC и др. [1]. В вопросе выбора критерия качества есть множество нюансов, которые в первую очередь обусловлены несбалансированностью как обучающих, так и тестирующих наборов данных по классам. Подробный анализ всего множества критериев эффективности решения задачи классификации не является целью настоящей работы, отметим лишь то, что в качестве критерия качества желательно выбирать тот критерий, который в максимальной степени отражает цели классификации для конкретной задачи. При этом желательно выбирать такой критерий, который мог быть полезен сразу для целой группы задач и при этом не зависел бы от локальных параметров классификатора, например, от уровня порога принятия решения. Но при всем этом достоверно характеризовал бы информативность классификатора.

Всем этим требованиям отвечает критерий AUC-ROC (Area Under the Curve ROC), где ROC — это так называемая Receiver Operator Characteristic [7]. В отличие от иных критериев качества классификации, величина AUC-ROC не зависит от порога принятия решения и адекватно характеризует общую эффективность (информативность) классификатора в следующем смысле: классификатор должен выделять как можно больше объектов целевого класса и как можно меньше объектов всех остальных классов. Формально это может быть записано следующим образом:

$$IC(f(\beta|h)|\Theta) = \frac{\left\langle\left\langle [f(\beta|h)(z^{(\theta)}_{tr}) \neq \theta]\right\rangle_{z^{(\theta)}_{tr} \in Z_T}\right\rangle_{\theta\in\Theta}}{\left\langle\left\langle [f(\beta|h)(z^{(\theta)}_{tr}) = \theta]\right\rangle_{z^{(\theta)}_{tr} \in Z_T}\right\rangle_{\theta\in\Theta}}.$$

Чем меньше величина $IC(f(\beta|h)|\Theta)$, тем более информативен классификатор $f(\beta|h)$ для заданных β и h. Наилучший результат соответствует ситуации $IC(f(\beta|h)|\Theta) \to \min\limits_{\beta,h}$.

Эффективность классификатора в мультиклассовой постановке задачи оценивается в рамках широко известной стратегии «one vs rest» (OvR or one-vs.-all, OvA or one-against-all, OAA) [8] путем сведения мультиклассовой задачи к серии бинарных задач: принадлежит ли образец z к классу θ, или он принадлежит к классу $\Theta\backslash\theta$, $\theta \in \Theta$.

В этой связи обозначим:

— $\forall\theta \in \Theta$: $P_\theta(T|f(\beta|h)(z) > T)$ — вероятность того, что при использовании классификатора $f(\beta|h)$ объект z принадлежит к классу θ, если решение принимается согласно правилу $f(\beta|h)(z) > T$;

— $\forall\theta \in \Theta$: $P_{\Theta\backslash\theta}(T|f(\beta|h)(z) > T)$ — вероятность того, что при использовании классификатора $f(\beta|h)$ объект z принадлежит классу $\Theta\backslash\theta$, если решение принимается согласно правилу $f(\beta|h)(z) > T$;

— $\mathbf{z}^{(\Theta\backslash\theta)}_{tr} \subset \bigcup_{q\in\Theta\backslash\theta} z^{(q)}_{tr}$;

— $z^{(\theta)}_{tr} \subset Z^{(\theta)}_{tr}$;

— $TPR_f(T|\theta,\beta,h) = \int\limits_T^\infty P_\theta(x|f(\beta|h)(z) > x)dx$, $\theta \in \Theta$;

— $\mathbf{1}[a < b] = \begin{cases} 1, \text{ if } a < b \\ 0, \text{ if } a \geq b \end{cases}$;

— $[]_M$ — операция математического округления.

В этом случае параметр AUC-ROC для классификатора $f(\beta|h)$ и класса $\theta \in \Theta$ определяется согласно выражению:

$$AUC_f(\theta,\beta,h) = \int\limits_{-\infty}^{\infty} TPR_f(T|\theta,\beta,h)\, P_{\Theta\backslash\theta}(T|f(\beta|h)(z) > T)dT, \quad \theta \in \Theta.$$

По всему множеству Θ параметр AUC-ROC определяется согласно выражению:

$$AUC_{f,\Theta}(\beta,h) = \langle AUC_f(\theta,\beta,h)\rangle_{\theta\in\Theta},\ |\Theta| > 2.$$

Для практических расчетов удобно использовать оценку этой величины, которая определяется согласно выражению [9]:

$$AUC_f{}^*(\theta,\beta,h,\mathbf{Z}_{tr}) = \sum_{t_0\in z^{(\theta)}_{tr}} \sum_{t_1\in \mathbf{z}^{(\Theta\backslash\theta)}_{tr}} \mathbf{1}[f(\beta|h)(t_0) < f(\beta|h)(t_1)](|z^{(\theta)}_{tr}||\mathbf{z}^{(\Theta\backslash\theta)}_{tr}|)^{-1}.$$

$$AUC_{f,\Theta}{}^*(\beta,h,\mathbf{Z}_{tr}) = \langle AUC_f{}^*(\theta,\beta,h)\rangle_{\theta\in\Theta},\ |\Theta| > 2,$$
$$AUC_{f,\Theta}{}^*(\beta,h) \in [0,1].$$

Таким образом, далее будем полагать, что

$$\Upsilon_f(\beta(h), h|\Theta, \mathbf{Z}_{tr}) = 1 - AUC_{f,\Theta}{}^*(\beta,h,\mathbf{Z}_{tr}).$$

Метод решения

Обозначим $\mathbf{\Phi}_f$: $H \to \Upsilon_f(\beta_f(h), h|\Theta, \mathbf{Z}_{tr})$. Предлагаемый подход состоит из двух этапов. На первом этапе строится непараметрическая регрессия Λ_f, которая аппроксимирует функцию $\mathbf{\Phi}_f$. А на втором этапе, используя процедуру Нелдера–Мида, находится глобальный минимум этой функции на параметрическом компакте H, аргументы которого и являются решением поставленной задачи. Рассмотрим эти этапы более подробно.

Пусть определено множество опорных точек $F_f(\mathbf{h}) = \{(\mathbf{\Phi}_f(h_i), h_i)|h_i \in \mathbf{h}\}$, где $\mathbf{h} = \{h|h \in H\}$, $|\mathbf{h}| < \infty$.

Регрессию $\Lambda_f(\mathbf{h})$ можно аппроксимировать различными непараметрическими методами. Из группы альтернатив наибольшую эффективность показали аппроксимации методом опорных векторов (SV-regression) [10] и Gradient Boosting Regression [11]. Каждая из этих аппроксимаций, в свою очередь, зависима от соответствующих гиперпараметров, которые также нуждаются в настройке. Но эта настройка требует существенно меньше вычислительных ресурсов в сравнении с ресурсами, необходимыми для определения $F_f(\mathbf{h})$. Так как результаты использования этих аппроксимаций имеют сравнимую точность, далее будет использована только аппроксимация методом опорных векторов $\Lambda_f^{(SV,\alpha^*)}()$, которая менее требовательна к вычислительным ресурсам, и имеет набор гиперпараметров меньшей размерности. Обозначим: $\alpha^* \in A$ — набор гиперпараметров; A — известный компакт. Как уже было упомянуто ранее, крайне высокую вычислительную стоимость имеет процедура построения множества $F_f(\mathbf{h})$ за счет того, что вычисление величины $\mathbf{\Phi}_f(h_i)$ в точках $h_i \in \mathbf{h}$ производится с использованием поисковой процедуры $S_f(h_i|B,\mathbf{Z}_{tr})$ на корпусе данных $\mathbf{Z}_{tr}$ большой мощности. В связи с этим желательно минимизировать величину $|\mathbf{h}|$. С другой стороны, после вычисления $F_f(\mathbf{h})$ оптимизация гиперпараметров аппроксимации регрессии α представляет собой низкостоимостную в вычислительном смысле процедуру, ввиду невысокой размерности $|\alpha|$, а самое главное, — ввиду малости величины $|\mathbf{h}|$, поэтому оптимизация α производится, например, полным перебором на заданной решетке $A_L \subseteq A$ с малым шагом или методом случайного поиска при большом числе повторов. Формально эту процедуру можно записать так:

$$\alpha^* = \mathrm{Arg}\inf_{\alpha\in A_L} \langle|\mathbf{\Phi}_f(h_i) - \Lambda_f^{(SV,\alpha)}(h_i)|^l\rangle_{h_i\in\mathbf{h}},$$

где $l \geq 1$.

Для априорно заданной величины $n = |\mathbf{h}|$ множество $\mathbf{h}$ выбирается или случайным образом или неслучайным, например, путем формирования обычной регулярной решетки. В результате имеем аппроксимацию для $\mathbf{\Phi}_f(h_i)$ в виде $\Lambda_f^{(SV,\alpha^*)}()$.

На следующем шаге решается основная задача:

$$h_n = \mathrm{Arg}\inf_{h\in H} \Lambda_f^{(SV,\alpha^*)}(h).$$

Обозначим символом $Ind = \{1, \ldots, U\}$ — множество индексов всех компонентов компакта $H = \prod_{i=1}^{U} <H>_i$, где $<H>_i$ — ограниченное множество значений компонента $<h>_i$ гиперпараметра h: $<h>_i \in <H>_i$. Примем: $Int \subseteq Ind$ подмножество, состоящее из всех индексов компонентов компакта H, которые представляют собой конечные множества целых чисел, т. е. $i \in Int \Rightarrow <H>_i \subseteq N$.

Сначала решается следующая задача:

$$h_{n,f}(R^U) = \mathrm{Arg}\inf_{h\in R^U} \Lambda_f^{(SV,\alpha^*)}(h), \qquad (1)$$

где R^U — U-мерное вещественное пространство. Так как функционал $\Lambda_f^{(SV,\alpha^*)}(h)$ негладкий и зашумленный, для решения задачи (1) используется практически эффективный метод Нелдера–Мида [12], называемый методом деформируемого многогранника, который не требует вычисления градиентов. Суть данного метода заключается в последовательном перемещении и деформировании симплекса (U-мерного тетраэдра) вокруг точки экстремума. Этот метод чувствителен к начальной точке и может «застрять» на локальном экстремуме. Для того чтобы этого избежать, процедура запускается многократно, из различных точек H. На практике достаточно запустить процедуру Нелдера–Мида для каждого элемента множества $F_f(\mathbf{h})$, используя их в качестве начальных точек, и выбрать то решение $h_{n,f}(R^U)$, которое доставит минимум $\Lambda_f^{(SV,\alpha^*)}(h)$. Как правило, многократный запуск процедуры такого рода недорог в вычислительном аспекте, так как величина $|\mathbf{h}|$ — невелика. После того, как получено промежуточное решение $h_{n,f}(R^U)$, необходимо спроектировать его на компакт H, с учетом того, что часть компонентов H являются целыми числами.

Проектирование производится при помощи проектора $\Pr(h_{n,f}(R^U)|H)$, который описывается следующим образом:

$$h_{n,f} = \Pr(h_{n,f}(R^U)|H): \overset{U}{\underset{i=1}{\forall}} :< \Pr(h_{n,f}(R^U)|H) >_i =$$
$$= \begin{cases} [k(< h_{n,f}(R^U) >_i, H)]_M, \; if\,(i \in Int) \\ k(< h_{n,f}(R^U) >_i, H), \; if\,(i \in (Ind \backslash Int) \end{cases} \qquad (2)$$
$$k(a,H) = \begin{cases} a, \; if\; a \cap H \neq \varnothing \\ p(a,H), \; if\; a \cap H = \varnothing \end{cases}, \; p(a,H) = \mathrm{Arg}\inf_{h\in H} \|a-h\|.$$

Оценка $h_{n,f}$, полученная в (2) является решением поставленной задачи. Качество полученного решения определяется величиной $\Upsilon_f(\beta_f(h_{n,f}), h_{n,f})$. Чем меньше эта величина, тем выше качество выбора набора гиперпараметров $h_{n,f}$. Формулы (1) и (2) совместно полностью определяют искомую процедуру $O_{n,f}(\mathbf{Z}_{tr}, \mathbf{h})$.

Методика тестирования

Предложенный метод выбора гиперпараметров тестировался для двух широко известных алгоритмов машинного обучения (Machine Learning): Multiclass SVM-classifier (MC SVM) и Multiclass Gradient Boosting Classifier (MC GB). Наборы использованных гиперпараметров:

— для **MC SVM**:

$$h_{SVM} = (C, \gamma, tol) \in H =$$
$$= [1,300] \otimes [0,1/\mathrm{Dim}(Z), 10/\mathrm{Dim}(Z)] \otimes [0{,}5\cdot 10^{-3}, 10^{-2}],$$

где $\mathrm{Dim}(Z)$ — размерность пространства признаков Z; γ — параметр ядерной функции (gamma); C — регуляризирующая константа; tol — допустимый критерий остановки;

— для **MC GB**:

$$h_{GB} = (lr, n\_e, ss, mss, msl, md, mln, mf, tol) \in H,$$
$$H = [0{,}01;\, 1] \otimes \{50, \ldots, 300\} \otimes [0{,}5;\, 3] \otimes \{2, \ldots, 4\} \otimes$$
$$\{1, \ldots 4\} \otimes \{2, \ldots, 6\} \otimes \{2, \ldots, 30\} \otimes \{\sqrt{\mathrm{Dim}(Z)}, \ldots,$$
$$\mathrm{Dim}(Z)\} \otimes [0{,}5\cdot 10^{-4}, 10^{-3}],$$

где lr — leaningrate; n_e — n_estimators; ss — subsample; mss — min_samples_split; md — max_depth; mln — max_leaf_nodes; mf — max_features; tol — допустимый критерий остановки.

Основным параметром алгоритма, который во многом определяет объем необходимых вычислительных ресурсов (стоимость решения задачи), является параметр $n = |\mathbf{h}|$ — число опорных точек $F_f(\mathbf{h}) = \{(\mathbf{\Phi}_f(h_i), h_i)|h_i \in \mathbf{h}\}$, необходимых для восстановления регрессии $\Lambda_f^{(SV;\alpha^*)}(h)$. Чем больше n, тем больше стоимость решения, так как вычисление величины $\mathbf{\Phi}_f(h)$ при фиксированном $h \in \mathbf{h}$ может представлять собой высокозатратную задачу, поэтому величина n в идеале подлежит минимизации.

Идея вычислительного эксперимента, цель которого — сравнение эффективности предложенного решения с двумя опорными (базовым и вычисленным по методу случайного поиска) для некого классификатора $f(\beta|h)$, может быть представлена в виде следующей последовательности шагов.

1. Для $DS_j \in \mathbf{DS}$, вычисляются множества: $\Psi_{n,j} = \{F_f^{(j)}(\mathbf{h}_n^{(i,j)})|i = 1, \ldots, p\}$, $n \in \mathbf{n} = \{10, 20, 30\}$, $\mathbf{h}_n^{(i,j)} = \{h|h \in H\}$, $|\mathbf{h}_n^{(i,j)}| = n$, $i = 1, \ldots, p$; $p = 30$. Здесь множества $\mathbf{h}_n^{(i,j)}$ определяются случайным образом.
2. Для каждого $F_f^{(j)}(\mathbf{h}_n^{(i,j)}) \in \Psi_{n,j}$, согласно (1) и (2) определяются локальные решения $h_{n,f}^{(i,j)}$ и формируется множество решений $\{h_{n,f}^{(i,j)})|i = 1, \ldots, p\}$, а также соответствующее ему множество оценок качества этих решений $\{\Upsilon_f(\beta_f(h_{n,f}^{(i,j)}), h_{n,f}^{(i,j)})|i = 1, \ldots, p\}$. Это самый затратный в вычислительном плане этап решения.
3. Элементы $\mathbf{h}_n^{(i,j)}|i = 1, \ldots, p$, используются как опорные точки для метода случайного поиска [2], т. е. $h_{ST,f,n}^{(i,j)} = \mathrm{Arg}\inf_{h \in \mathbf{h}_n^{(i,j)}} \Upsilon_f(\beta(h), h)$.
4. Для каждого $n \in \mathbf{n}$, элемента множества $\{h_{n,f}^{(i,j)})|i = 1, \ldots, p\}$, а также $DS_j \in \mathbf{DS}$ вычисляются простые сравнительные показатели эффективности предложенного алгоритма по сравнению с двумя опорными решениями ($h_B^{(i,j)}$ и $h_{ST}^{(i,j)}$) в виде:

$$\varepsilon_{B,f}^{(j,i)}(n) = 100(\Upsilon_f(\beta(h_{B,f}), h_{B,f}) - \Upsilon_f(\beta(h_{n,f}^{(i,j)}), h_{n,f}^{(i,j)})/\Upsilon_f(\beta(h_{B,f}), h_{B,f}),$$

$$\varepsilon_{ST,f}^{(j,i)}(n) = 100(\Upsilon_f(\beta(h_{ST,f,n}^{(i,j)}), h_{ST,f,n}^{(i,j)}) - \Upsilon_f(\beta(h_{n,f}^{(i,j)}), h_{n,f}^{(i,j)})/\Upsilon_f(\beta(h_{ST,f,n}^{(i,j)}), h_{ST,f,n}^{(i,j)}).$$

Таким образом, формируются множества $\{\varepsilon_{B,f}^{(j,i)}(n)|i = 1, \ldots, p\}$ и $\{\varepsilon_{ST,B,f}^{(j,i)}(n)|i = 1, \ldots, p\}$. Показатель $\varepsilon_{B,f}^{(j,i)}$ — относительное (в процентах) улучшение показателя качества предложенного алгоритма по сравнению с базовым для набора $n \in \mathbf{n}$, DS_j и $h_{n,f}^{(i,j)}$. Соответственно $\{\varepsilon_{ST,B,f}^{(i,j)}$— аналогичный показатель при сравнении предложенного метода с методом случайного поиска.

Согласно методике [13], для каждых $DS_j \in \mathbf{DS}$, $n \in \mathbf{n}$ и $\{\varepsilon_{B,f}^{(i,j)}(n)|i = 1, \ldots, p\}$, строятся байесовские доверительные интервалы уровня 0,9: $CI_{n,j}^{ST} \subseteq H$ и $CI_{n,j}^{B} \subseteq H$ для величин $\mathbf{E}\varepsilon_{ST,f}^{(j)}(n)$ и $\mathbf{E}\varepsilon_{B,f}^{(j)}(n)$ соответственно, где $\mathbf{E}\chi$ — математическое ожидание величины χ. Тогда имеем: $\mathbf{P}(\mathbf{E}\varepsilon_{ST,f}^{(j)}(n) \in CI_{n,j}^{ST}) \geq 0{,}9$ и $\mathbf{P}(\mathbf{E}\varepsilon_{B,f}^{(j)}(n) \in CI_{n,j}^{ST}) \geq 0{,}9$. Ширина этих доверительных интервалов характеризует точность оценивания.

Корпуса данных, использованные для численных экспериментов

Для тестирования использовались корпуса IRIS, DIGITS и SS@, совместно образующие множество $\mathbf{DS} = \{DS_j|j = 1{,}3\}$. Корпуса IRIS и DIGITS являются стандартными и доступны через среду Python, а SS@ — корпус данных C\F-OTDR сигналов от сейсмоакустических событий, собранный на нескольких полигонах в разное время года. Краткие спецификации корпусов:

— DIGITS — 1 797 изображений 8 × 8, рукописный текст; 10 классов, каждый класс из 180 образцов; размерность пространства признаков — 64;
— IRIS — параметры лепестков цветов, 3 класса; каждый класс из 50 образцов; размерность пространства признаков — 4;
— SS@ — сейсмоакустические сигналы; 7 классов; количество образцов по классам (101, 50, 124, 145, 150, 150, 200); размерность пространства признаков 24.

Результаты тестирования

В таблице приведены результаты численного эксперимента для $n = 10$ и корпусов данных **DS**.

Таблица. Результаты численного эксперимента

Алгоритм	Метод	Показатель	Базы данных		
			IRIS	DIGITS	SS@
MC SVM	Предложенный	$\overline{\Upsilon_f(n,j)}$*	$1{,}320 \cdot 10^{-4}$	$3{,}944 \cdot 10^{-4}$	$5{,}400 \cdot 10^{-3}$
	Базовый	$\Upsilon_f(\beta(h_{B,f}), h_{B,f})$	$2{,}200 \cdot 10^{-4}$	$5{,}800 \cdot 10^{-4}$	$3{,}000 \cdot 10^{-2}$
	Случайный	$\overline{\Upsilon_f(ST,n,j)}$**	$1{,}435 \cdot 10^{-4}$	$4{,}287 \cdot 10^{-4}$	$7{,}606 \cdot 10^{-3}$
MC GBR	Предложенный	$\overline{\Upsilon_f(n,j)}$*	$2{,}024 \cdot 10^{-4}$	$6{,}320 \cdot 10^{-4}$	$1{,}750 \cdot 10^{-2}$
	Базовый	$\Upsilon_f(\beta(h_{B,f}), h_{B,f})$	$8{,}800 \cdot 10^{-4}$	$7{,}900 \cdot 10^{-4}$	$2{,}500 \cdot 10^{-2}$
	Случайный	$\overline{\Upsilon_f(ST,n,j)}$**	$2{,}300 \cdot 10^{-4}$	$6{,}653 \cdot 10^{-4}$	$1{,}902 \cdot 10^{-2}$

* $\overline{\Upsilon_f(n,j)} = \sum_{i=1}^{p} \Upsilon_f(\beta(h_{n,f}^{(i,j)}), h_{n,f}^{(i,j)})p^{-1}$;

** $\overline{\Upsilon_f(ST,n,j)} = \sum_{i=1}^{p} \Upsilon_f(\beta(h_{ST,f,n}^{(i,j)}), h_{ST,f,n}^{(i,j)})p^{-1}$.

Из таблицы можно видеть, что требования постановки задачи выполнены. Более удобна демонстрация преимущества предложенного метода при переходе к сравнительным (процентным) показателям $\mathbf{E}\varepsilon^{(j)}_{ST,f}(n)$ и $\mathbf{E}\varepsilon^{(j)}_{B,f}(n)$.

На рис. 1–3 в графическом виде представлены результаты численного моделирования для этих показателей по всем корпусам $\mathbf{DS} = \{DS_j | j = 1,3\}$ и множеству $\mathbf{n} = \{10, 20, 30\}$. По оси ординат отложены величины $\mathbf{E}\varepsilon^{(j)}_{ST,f}(n)$ (или $\mathbf{E}\varepsilon^{(j)}_{B,f}(n)$) и 90 % доверительные интервалы для них. А по оси абсцисс — параметр n (число пробных вычислений функционала качества Υ_f для различных значений гиперпараметра).

Названия графиков включают следующую последовательность тэгов: <Имя Базы> /<Алгоритм>/<Тип>. Тэг <Имя Базы> принимает значения имен тестовых баз из множества **DS**. Тэг <Алгоритм> принимает два значения: MC SVM или MC GBR, в зависимости от того, какой алгоритм классификации исследовался в данном эксперименте. Значение тэга <Тип> обозначает: метод оптимизации гиперпараметров, с которым сравнивается предложенный метод в данном эксперименте: с опорным решением (Basic) или с методом случайного поиска (Random).

Представленные результаты демонстрируют, что все требования поставленной задачи достигнуты. Ширина доверительного интервала в каждом эксперименте предсказуемо уменьшается с увеличением величины n. Предложенный метод оптимизации гиперпараметров, проверенный для двух типов классификаторов, обеспечивает значительный относительный прирост показателя качества Υ_f при сравнении с опорным решением, и его величина лежит в интервале 20–80 %. Но и в случае сравнения с методом случайного поиска, относительный прирост качества достаточно значим, находясь в интервале 5–28 %.

Заключение

Предложенный метод выбора гиперпараметров основан на идее о возможности гладкой аппроксимации функции значений критерия качества (значения регрессии) в зависимости от величин гиперпараметров (регрессоров). В этом случае, используя специальные методы поиска экстремума в многомерном пространстве, появляется возможность вычислять такие значения гиперпараметров, которые не лежат в узлах предварительно вычисленной сетки (пробного множества), но при этом доставляют сравнительно лучшие значения критерия качества по сравнению с узловыми (пробными) значениями гиперпараметров. Численные эксперименты, проведенные для двух типов классификаторов в мультиклассовой постановке на трех базах данных, показали, что предложенная идея вполне работоспособна и дает выигрыш от 8 до 80 % в зависимости от тестовой базы, опорного решения и типа классификатора.

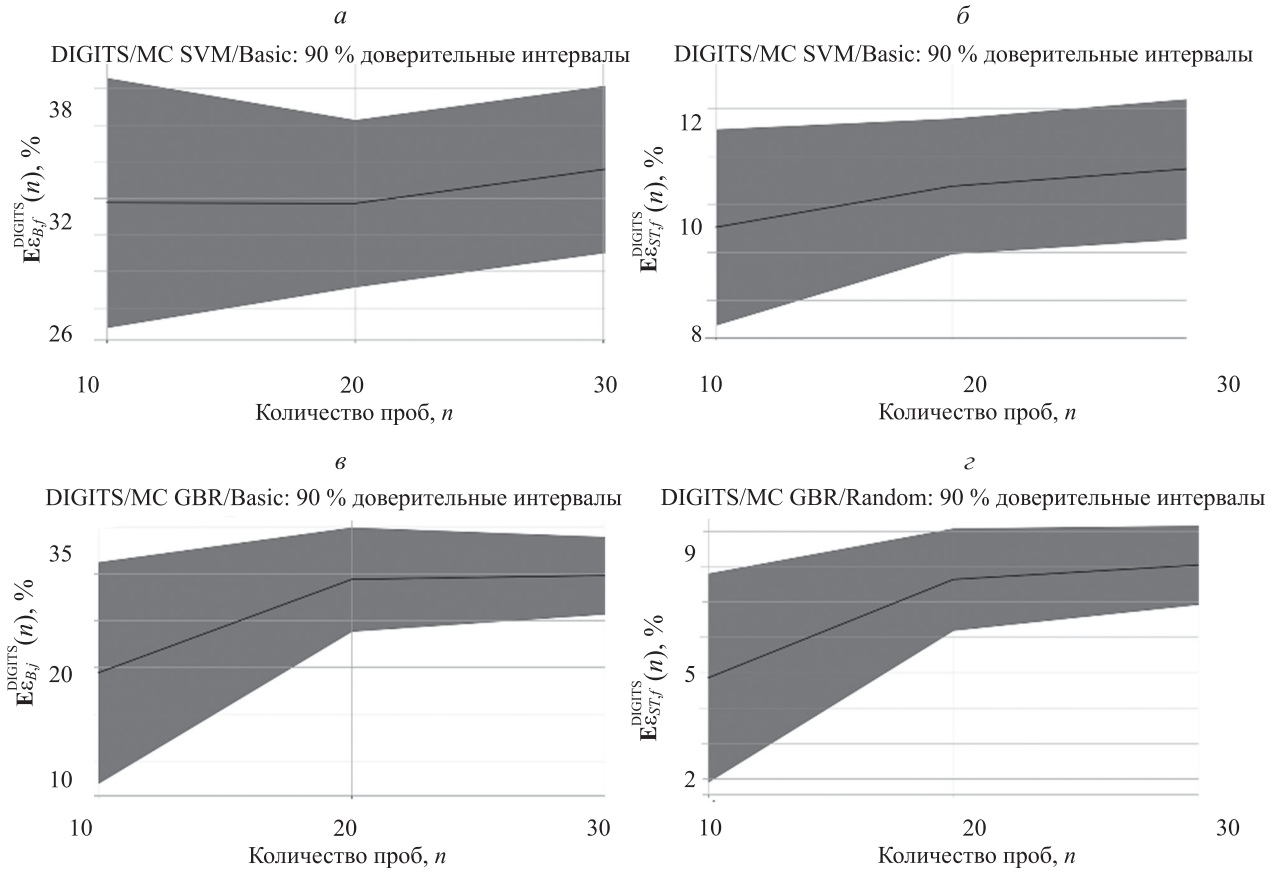

Рис. 1. Результаты экспериментов на корпусе DIGITS

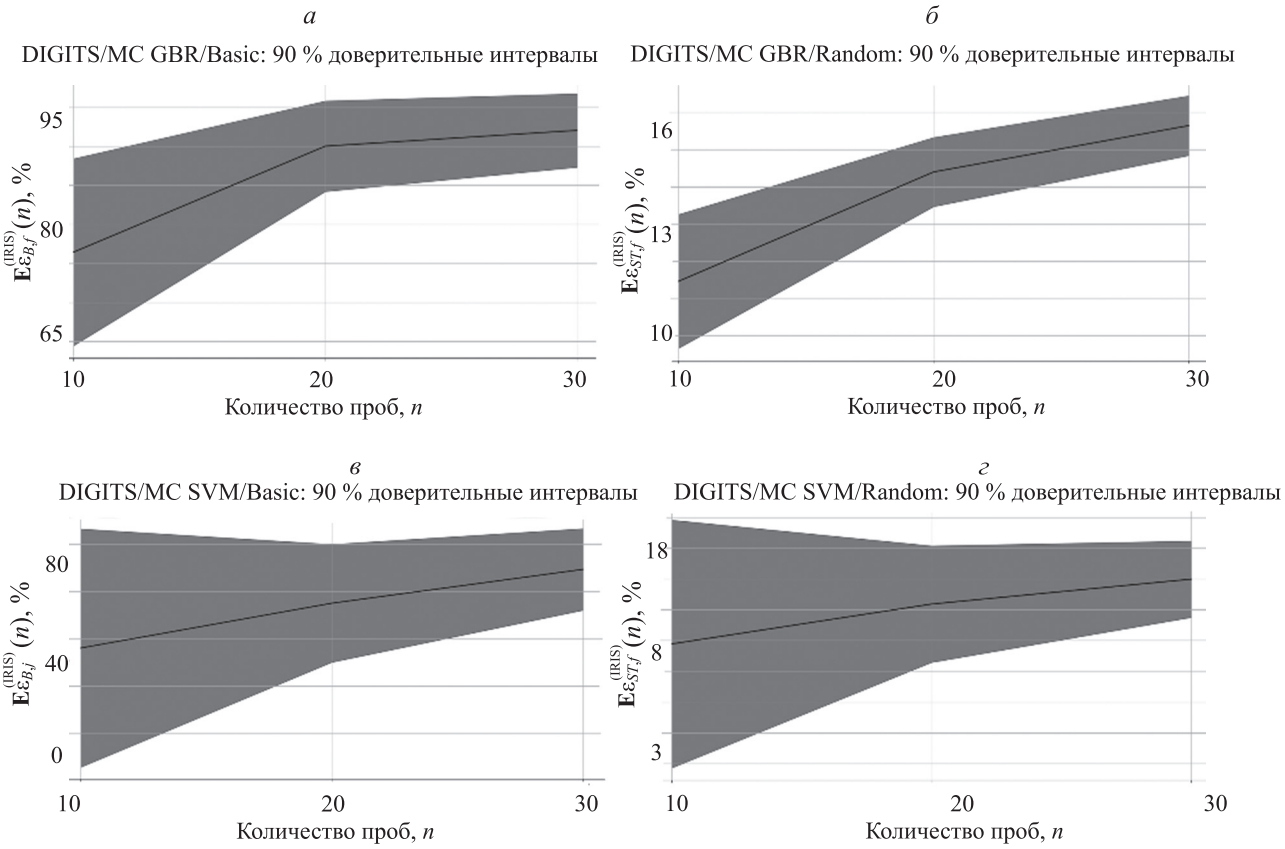

Рис. 2. Результаты экспериментов на корпусе IRIS

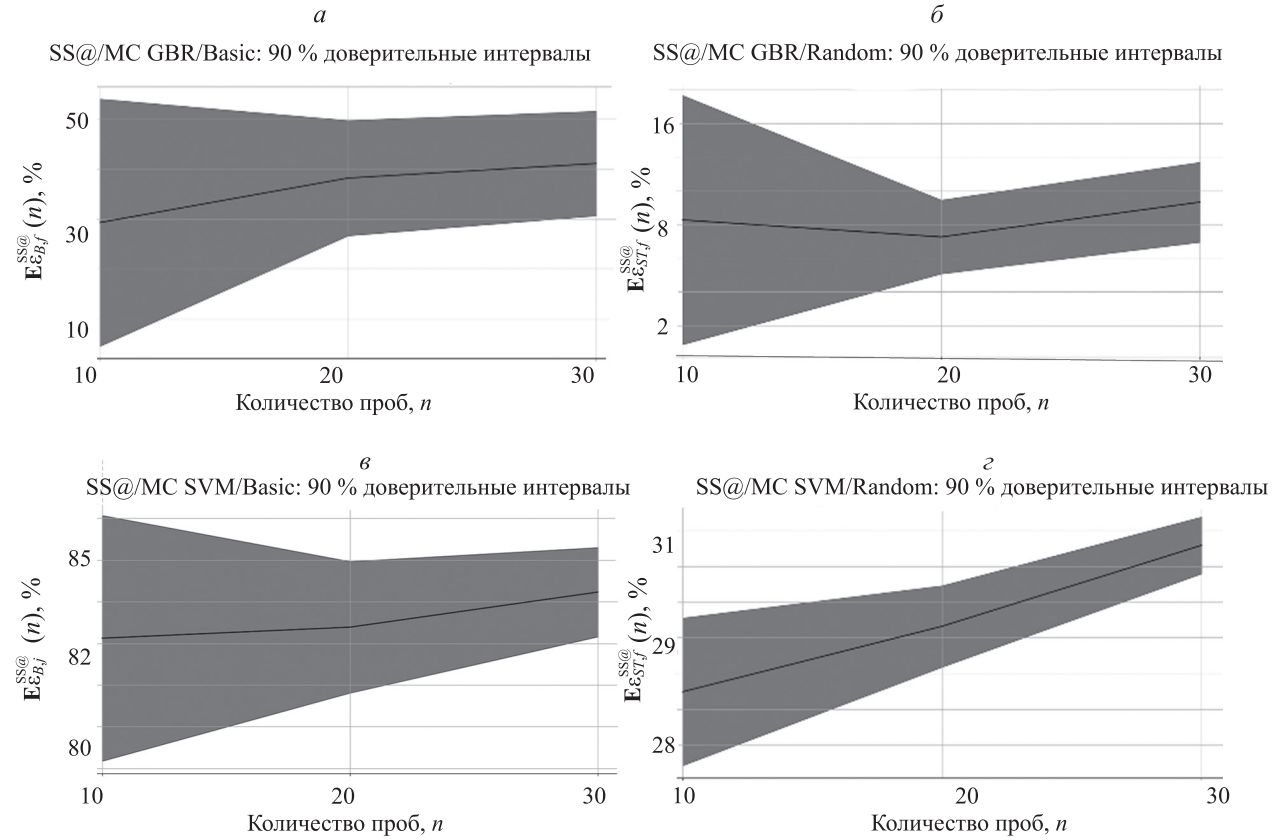

Рис. 3. Результаты экспериментов на корпусе SS@

Литература

1. Montgomery D.C. Design and Analysis of Experiments. 8th ed. John Wiley & Sons, 2013. 752 p.
2. Bergstra J., Bengio Y. Random search for hyper-parameter optimization // Journal of Machine Learning Research. 2012. V. 13. P. 281–305.
3. Zeng X., Luo G. Progressive sampling-based Bayesian optimization for efficient and automatic machine learning model selection // Health Information Science and Systems. 2017. V. 5. P. 2. doi: 10.1007/s13755-017-0023-z
4. Zhang Y., Bahadori M.T., Su H., Sun J. FLASH: Fast bayesian optimization for data analytic pipelines // Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). 2016. P. 2065–2074. doi: 10.1145/2939672.2939829
5. Rasmussen C., Williams C. Gaussian Processes for Machine Learning. The MIT Press, 2006. 248 p.
6. Maclaurin D., Duvenaud D., Adams R. Gradient-based hyperparameter optimization through reversible learning // ICML'15: Proc. of the 32nd International Conference on International Conference on Machine Learning. 2015. P. 2113–2122.
7. Powers D.M. Evaluation: from precision, recall and F-measure to ROC, Informedness, markedness & correlation // Journal of Machine Learning Technologies. 2011. V. 2. N 1. P. 37–63.
8. Bishop C.M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
9. Calders T., Jaroszewicz S. Efficient AUC optimization for classification // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2007. V. 4702. P. 42–53. doi: 10.1007/978-3-540-74976-9_8
10. Drucker H., Burges C.J.C., Kaufman L., Smola A., Vapnik V. Support vector regression machines // Advances in Neural Information Processing Systems. 1997. V. 9. P. 155–161.
11. Friedman J.H. Greedy function approximation: A gradient boosting machine // Annals of Statistics. 2001. V. 29. N 5. P. 1189–1232. doi: 10.1214/aos/1013203451
12. Nelder J.A., Mead R. A simplex method for function minimization // Computer Journal. 1965. V. 7. N 4. P. 308–313. doi: 10.1093/comjnl/7.4.308
13. Oliphant T.E. A Bayesian perspective on estimating mean, variance, and standard-deviation from data [Электронный ресурс]. URL: https://scholarsarchive.byu.edu/facpub/278 (дата обращения: 04.06.20).

Авторы

Тимофеев Андрей Владимирович — доктор технических наук, научный директор, ТОО «Эквалайзум», Астана, 010000, Казахстан, Scopus ID: 56689367600, ORCID ID: 0000-0001-7212-5230, timofeev.andrey@gmail.com

References

1. Montgomery D.C. *Design and Analysis of Experiments*. 8th ed. John Wiley & Sons, 2013, 752 p.
2. Bergstra J., Bengio Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 2012, vol. 13, pp. 281–305.
3. Zeng X., Luo G. Progressive sampling-based Bayesian optimization for efficient and automatic machine learning model selection. *Health Information Science and Systems*, 2017, vol. 5, pp. 2. doi: 10.1007/s13755-017-0023-z
4. Zhang Y., Bahadori M.T., Su H., Sun J. FLASH: Fast bayesian optimization for data analytic pipelines. *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016, pp. 2065–2074. doi: 10.1145/2939672.2939829
5. Rasmussen C., Williams C. *Gaussian Processes for Machine Learning*. The MIT Press, 2006, 248 p.
6. Maclaurin D., Duvenaud D., Adams R. Gradient-based hyperparameter optimization through reversible learning. *ICML'15: Proc. of the 32nd International Conference on International Conference on Machine Learning*, 2015, pp. 2113–2122.
7. Powers D.M. Evaluation: from precision, recall and F-measure to ROC, Informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2011, vol. 2, no. 1, pp. 37–63.
8. Bishop C.M. *Pattern Recognition and Machine Learning*. Springer, 2006, 738 p.
9. Calders T., Jaroszewicz S. Efficient AUC optimization for classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4702, pp. 42–53. doi: 10.1007/978-3-540-74976-9_8
10. Drucker H., Burges C.J.C., Kaufman L., Smola A., Vapnik V. Support vector regression machines. *Advances in Neural Information Processing Systems*, 1997, vol. 9, pp. 155–161.
11. Friedman J.H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 2001, vol. 29, no. 5, pp. 1189–1232. doi: 10.1214/aos/1013203451
12. Nelder J.A., Mead R. A simplex method for function minimization. *Computer Journal*, 1965, vol. 7, no. 4, pp. 308–313. doi: 10.1093/comjnl/7.4.308
13. Oliphant T.E. A *Bayesian perspective on estimating mean, variance, and standard-deviation from data*. Available at: https://scholarsarchive.byu.edu/facpub/278 (accessed: 04.06.20).

Authors

Andrey V. Timofeev — D.Sc., Chief Scientific Officer, LLP EqualiZoom, Astana, 010000, Republic of Kazakhstan, Scopus ID: 56689367600, ORCID ID: 0000-0001-7212-5230, timofeev.andrey@gmail.com