КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2022 Т. 14 № 6 С. 1221–1238
DOI: 10.20537/2076-7633-2022-14-6-1221-1238

Ки&М

ЧИСЛЕННЫЕ МЕТОДЫ И ОСНОВЫ ИХ РЕАЛИЗАЦИИ

УДК: 519.8

Свойства алгоритмов поиска оптимальных порогов для задач многозначной классификации

А. И. Бергер[a], С. А. Гуда[b]

Южный федеральный университет,
Россия, 344006, г. Ростов-на-Дону, ул. Большая Садовая, 105/42

E-mail: [a] anna.ig.berger@gmail.com, [b] gudasergey@gmail.com

Получено 24.02.2022, после доработки — 09.06.2022.
Принято к публикации 08.09.2022.

Модели многозначной классификации возникают в различных сферах современной жизни, что объясняется всё большим количеством информации, требующей оперативного анализа. Одним из математических методов решения этой задачи является модульный метод, на первом этапе которого для каждого класса строится некоторая ранжирующая функция, упорядочивающая некоторым образом все объекты, а на втором этапе для каждого класса выбирается оптимальное значение порога, объекты с одной стороны которого относят к текущему классу, а с другой — нет. Пороги подбираются так, чтобы максимизировать целевую метрику качества. Алгоритмы, свойства которых изучаются в настоящей статье, посвящены второму этапу модульного подхода — выбору оптимального вектора порогов. Этот этап становится нетривиальным в случае использования в качестве целевой метрики качества F-меры от средней точности и полноты, так как она не допускает независимую оптимизацию порога в каждом классе. В задачах экстремальной многозначной классификации число классов может достигать сотен тысяч, поэтому исходная оптимизационная задача сводится к задаче поиска неподвижной точки специальным образом введенного отображения $\boldsymbol{V}$, определенного на единичном квадрате на плоскости средней точности P и полноты R. Используя это отображение, для оптимизации предлагаются два алгоритма: метод линеаризации F-меры и метод анализа области определения отображения $\boldsymbol{V}$. На наборах данных многозначной классификации разного размера и природы исследуются свойства алгоритмов, в частности зависимость погрешности от числа классов, от параметра F-меры и от внутренних параметров методов. Обнаружена особенность работы обоих алгоритмов для задач с областью определения отображения $\boldsymbol{V}$, содержащей протяженные линейные участки границ. В случае когда оптимальная точка расположена в окрестности этих участков, погрешности обоих методов не уменьшаются с увеличением количества классов. При этом метод линеаризации достаточно точно определяет аргумент оптимальной точки, а метод анализа области определения отображения $\boldsymbol{V}$ — полярный радиус.

Ключевые слова: многозначная классификация, экстремальная классификация, F-мера, метод линеаризации, метод анализа области определения

Работа выполнена при финансовой поддержке Российского научного фонда (проект 20-43-01015).

COMPUTER RESEARCH AND MODELING
2022 VOL. 14 NO. 6 P. 1221–1238
DOI: 10.20537/2076-7633-2022-14-6-1221-1238

Ки&М

NUMERICAL METHODS AND THE BASIS FOR THEIR APPLICATION

UDC: 519.8

Optimal threshold selection algorithms for multi-label classification: property study

A. I. Berger[a], S. A. Guda[b]

Southern Federal University,
105/42 Bolshaya Sadovaya st., Rostov-on-Don, 344006, Russia

E-mail: [a] anna.ig.berger@gmail.com, [b] gudasergey@gmail.com

Received 24.02.2022, after completion — 09.06.2022.
Accepted for publication 08.09.2022.

Multi-label classification models arise in various areas of life, which is explained by an increasing amount of information that requires prompt analysis. One of the mathematical methods for solving this problem is a plug-in approach, at the first stage of which, for each class, a certain ranking function is built, ordering all objects in some way, and at the second stage, the optimal thresholds are selected, the objects on one side of which are assigned to the current class, and on the other — to the other. Thresholds are chosen to maximize the target quality measure. The algorithms which properties are investigated in this article are devoted to the second stage of the plug-in approach which is the choice of the optimal threshold vector. This step becomes non-trivial if the F-measure of average precision and recall is used as the target quality assessment since it does not allow independent threshold optimization in each class. In problems of extreme multi-label classification, the number of classes can reach hundreds of thousands, so the original optimization problem is reduced to the problem of searching a fixed point of a specially introduced transformation $\boldsymbol{V}$, defined on a unit square on the plane of average precision P and recall R. Using this transformation, two algorithms are proposed for optimization: the F-measure linearization method and the method of $\boldsymbol{V}$ domain analysis. The properties of algorithms are studied when applied to multi-label classification data sets of various sizes and origin, in particular, the dependence of the error on the number of classes, on the F-measure parameter, and on the internal parameters of methods under study. The peculiarity of both algorithms work when used for problems with the domain of $\boldsymbol{V}$, containing large linear boundaries, was found. In case when the optimal point is located in the vicinity of these boundaries, the errors of both methods do not decrease with an increase in the number of classes. In this case, the linearization method quite accurately determines the argument of the optimal point, while the method of $\boldsymbol{V}$ domain analysis — the polar radius.

Keywords: multi-label classification, extreme classification, F-measure, linearization method, domain analysis method

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 6, pp. 1221–1238 (Russian).

The authors acknowledge Russian Science Foundation grant No. 20-43-01015 for the financial support.

1. Введение

Одной из проблем, с которой сталкиваются компании по всему миру, является существенный объем текстовой и визуальной информации, которая, с одной стороны, требует незамедлительного анализа, а с другой — не может быть обработана вручную ввиду своего значительного размера. Так возникает задача отнесения объектов к одной или нескольким категориям — задача классификации. Классификационные модели появляются в различных областях, таких как, например, медицина [Данилов и др., 2020], транспорт [Сабиров, Катасёв, Дагаева, 2021], банковское дело [Орлова, 2013], обработка естественного языка [Lagutina et al., 2014] и многих других.

На практике часто оказывается, что объект принадлежит сразу нескольким классам, что приводит к созданию моделей многозначной классификации, а если число классов велико — экстремальной многозначной классификации. В совокупности с большим количеством классифицируемых объектов это делает актуальной задачу создания таких математических методов, которые позволят найти оптимальное в смысле выбранной метрики качества решение наиболее эффективно.

Задачу многозначной классификации можно решать в два этапа. На первом этапе некоторая ранжирующая функция $\boldsymbol{\eta}\colon \mathbb{X} \to [0;\,1]^n$ упорядочивает объекты в каждом классе (n — количество классов). На втором этапе определяется вектор оптимальных порогов $\boldsymbol{T} = (t_1, t_2, \ldots, t_n)$, задающий для каждого упорядоченного списка число объектов t_k из начала, которые будут отнесены к классу так, чтобы максимизировать значение выбранной метрики качества. Такой подход называется модульным. Он был исследован в работах [Yang, 2001; Pillai, Fumera, Roli, 2013; Koyejo et al., 2015]. Среди достоинств порогового классификатора можно выделить его состоятельность в смысле сходимости к оптимальному байесовскому классификатору при условии стремления оценок принадлежности объектов классам к истинным вероятностям [Dembczynski, Jachnik, 2013].

Одной из широко используемых метрик качества в задачах многозначной классификации является F-мера (см. [Rijsbergen, 1979]), являющаяся гармоническим средним точности и полноты. Для обеспечения равного вклада в метрику классов разного размера используется макроусреднение, которое может производиться двумя способами, отличающимися порядком применения операций усреднения по классам и расчета гармонического среднего точности и полноты. Усреднение значений F-меры, вычисленных в каждом классе, macro-F является более простым с точки зрения оптимизации, так как позволяет находить оптимальный порог независимо в каждом классе. Эта метрика используется в таких работах, как [Fan, Lin, 2007; Pillai, Fumera, Roli, 2017; Lipton, Elkan, Naryanaswamy, 2014; Decubber et al., 2013]. В работах [Cornolti, Ferragina, Ciaramita, 2018; Luo, Li, 2014; Tran et al., 2018] находит применение вычисление F-меры другим способом, который мы будем обозначать как F-macro. Он заключается в вычислении гармонического среднего от усредненной по классам точности P и полноты R. При оптимизации метрики F-macro выбор порогов в одном классе влияет на выбор порогов в другом. Алгоритмы, свойства которых исследуются в настоящей статье, оптимизируют второй, более сложный вариант метрики качества, а также его обобщение — метрику F_β-macro, которая позволяет регулировать вклад точности и полноты в финальное значение меры.

В работе [Berger, Guda, 2020] произведено сведение задачи оптимизации F-меры от средних точности P и полноты R к задаче поиска неподвижной точки (P, R) некоторого специальным образом введенного отображения $\boldsymbol{V}$ на квадрате $[0;\,1]^2$ и предложены несколько алгоритмов поиска неподвижной точки для решения задач экстремальной многозначной классификации. Они основаны на свойствах отображения $\boldsymbol{V}$, построенного для линеаризованной F-меры, поэтому здесь и далее мы будем называть этот подход методом линеаризации. Статья [Berger, Guda, 2021] посвящена другой группе алгоритмов определения оптимального вектора порогов, основанных на анализе области определения отображения $\boldsymbol{V}$.

Целью данной работы является исследование погрешностей предложенных алгоритмов в зависимости от числа классов и внутренних параметров алгоритмов (таких как, например, число направлений в методе анализа области определения). Для изучения свойств алгоритмов были выбраны три набора данных, принадлежащих различным предметным областям: изображениям (ESP Game, 20 770 изображений, 268 классов), документам (WikiLSHTC-325K — стандартный набор экстремальной многозначной классификации, 2 365 435 документов, 325 056 классов), и искусственно сгенерированный набор произвольных данных Example, призванный проиллюстрировать особенности работы предложенных алгоритмов в вырожденных случаях.

Статья организована следующим образом. § 2 посвящен постановке задачи многозначной классификации и введению используемой в качестве целевой метрики F-macro. В § 3 описываются два метода поиска оптимальных порогов, предложенные авторами в предыдущих работах, свойства которых детально изучаются в § 4. Подпараграф 4.1 посвящен описанию свойств используемых датасетов. Вариация количества используемых классов n позволяет в подпараграфе 4.2 построить зависимость от n погрешности определения оптимальных значений средней точности P, полноты R, а также полярного угла и радиуса точки (P, R). Исследование зависимости погрешности от параметра β меры F_β-macro позволяет установить в подпараграфе 4.3 антисимметричные свойства двух предложенных алгоритмов: метод линеаризации точнее всего определяет полярный угол оптимальной точки (P, R), а метод анализа области определения $\boldsymbol{V}$ — полярный радиус. В подпараграфе 4.4 исследуется зависимость алгоритма анализа области определения $\boldsymbol{V}$ от количества используемых направлений.

2. Постановка задачи

Рассмотрим модель многозначной классификации объектов из множества $\mathbb{X}$ на n возможно пересекающихся классов, и пусть $\mathbb{Y} = \{0, 1\}^n$ — множество возможных меток классов (1 означает, что объект принадлежит классу, 0 — нет). Эта задача может быть рассмотрена как в классической, так и в вероятностной формулировке. В первой формулировке классы c_k являются возможно пересекающимися подмножествами $\mathbb{X}$, и для $\boldsymbol{x} \in \mathbb{X}$ координаты метки $\boldsymbol{y}$ определяются формулой[1] $y_k \stackrel{def}{=} [\![\boldsymbol{x} \in c_k]\!]$. Многозначным классификатором будем называть отображение $\boldsymbol{h}\colon \mathbb{X} \to \mathbb{Y}$, а в качестве обучающей выборки будем использовать множество пар $\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right)_{i=1}^{m}$. Целью задачи в этом случае является построение классификатора $\boldsymbol{h}$, наилучшим образом аппроксимирующего точные значения $\boldsymbol{y}$.

В вероятностной постановке задачи предполагается, что пары $(\boldsymbol{x}, \boldsymbol{y})$ выбраны согласно некоторому неизвестному совместному вероятностному распределению $\mathbb{P}$, определенному на декартовом произведении $\mathbb{X}\times\mathbb{Y}$. Для заданного $\boldsymbol{x}$ значение $\boldsymbol{y}$ целевой функции является случайным. Эта постановка позволяет сформулировать задачу поиска оптимального классификатора как восстановление условной плотности вероятности $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$.

Для оценки качества классификатора с точки зрения вероятностного подхода используются такие метрики, как точность и полнота, характеризующие этот классификатор с разных сторон. Точность классификатора — условная вероятность объекта быть правильно классифицированным, то есть принадлежать классу при условии того, что классификатор отнес его к этому классу: $p_k(h_k) = \mathbb{P}(y_k = 1 \mid h_k = 1)$. Среди множества пороговых классификаторов, относящих к классу первые несколько объектов ряда, отсортированного по убыванию вероятности принадлежности, максимальную точность будет иметь первое положение порога, то есть классификатор, равный 1 только для первого, самого вероятного, объекта из отсортированного ряда.

Полнота — условная вероятность объекта быть отнесенным к классу, если объект действительно ему принадлежит: $r_k(h_k) = \mathbb{P}(h_k = 1 \mid y_k = 1)$. Полнота будет максимальной, в частности,

[1] Скобками $[\![\cdot]\!]$ здесь обозначен функционал, возвращающий 1, если условие истинно, и 0 — когда ложно.

в том случае, когда классификатор относит к классу все объекты множества $\mathbb{X}$, чему соответствует последнее положение порога.

Чтобы найти компромисс между этими двумя противоположными оценками, используют F-меру, представляющую собой гармоническое среднее между точностью и полнотой: $F = \frac{2pr}{p+r}$. Но в условиях многозначной классификации расчет метрики может происходить двумя способами, отличающимися порядком применения операций усреднения по классам и гармонического среднего. Первый способ заключается в усреднении значений F-меры, вычисленных независимо в каждом классе:

$$\text{macro-F}(\boldsymbol{h}) = \frac{1}{n}\sum_{k=1}^{n} F(p_k(h_k),\, r_k(h_k)). \tag{1}$$

Другой вариант F-меры, исследуемый в данной работе, состоит в расчете среднего гармонического средних точности и полноты:

$$P(\boldsymbol{h}) = \frac{1}{n}\sum_{k=1}^{n} p_k(h_k), \quad R(\boldsymbol{h}) = \frac{1}{n}\sum_{k=1}^{n} r_k(h_k), \quad \text{F-macro}(\boldsymbol{h}) = F(P(\boldsymbol{h}),\, R(\boldsymbol{h})) = \frac{2P(\boldsymbol{h})R(\boldsymbol{h})}{P(\boldsymbol{h}) + R(\boldsymbol{h})}. \tag{2}$$

Аналогично мере F-macro, которая по своей природе симметрична относительно точности и полноты, можно ввести меру F_β-macro, которая придает точности и полноте различный вес, что может быть удобно при решении некоторых реальных задач многозначной классификации:

$$\text{F}_\beta\text{-macro}(\boldsymbol{h}) = F_\beta(P(\boldsymbol{h}),\, R(\boldsymbol{h})) = \frac{\left(1+\beta^2\right)P(\boldsymbol{h})R(\boldsymbol{h})}{P(\boldsymbol{h}) + \beta^2 R(\boldsymbol{h})}. \tag{3}$$

При $0 < \beta < 1$ приоритет при оптимизации отдается точности, при $\beta > 1$ — полноте. F-macro является частным случаем F_β-macro при $\beta = 1$.

Метрика macro-F является более простой с точки зрения оптимизации, так как допускает независимую оптимизацию порога в каждом классе, в то время как при оптимизации F-macro маленькая полнота в одних классах может быть компенсирована высокой точностью в других, что не позволяет разделить оптимизацию на n независимых задач и делает ее вычислительно трудной.

Более того, в реальности мы ограничены одной тестовой выборкой, поэтому точные вероятности принадлежности классам неизвестны. Пусть $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$ — множество объектов и точных значений соответствующих им классов, $y_{ik} = [\![\boldsymbol{x}_i \in c_k]\!]$. Эмпирические варианты $\text{F-macro}^{\text{EM}}$ и $\text{macro-F}^{\text{EM}}$ вычисляются как оценки приведенных выше метрик с использованием следующих величин:

$$p_k^{\text{EM}}(h_k) = \frac{\sum\limits_{i=1}^{m} y_{ik} h_k(\boldsymbol{x}_i)}{\sum\limits_{i=1}^{m} h_k(\boldsymbol{x}_i)}, \qquad r_k^{\text{EM}}(h_k) = \frac{\sum\limits_{i=1}^{m} y_{ik} h_k(\boldsymbol{x}_i)}{\sum\limits_{i=1}^{m} y_{ik}}, \tag{4}$$

$$P^{\text{EM}}(\boldsymbol{h}) = \frac{1}{n}\sum_{k=1}^{n} p_k^{\text{EM}}(h_k), \quad R^{\text{EM}}(\boldsymbol{h}) = \frac{1}{n}\sum_{k=1}^{n} r_k^{\text{EM}}(h_k). \tag{5}$$

Мера F-macro монотонна по p_k и r_k, следовательно (см. [Berger, Guda, 2020]), оптимальный классификатор $\boldsymbol{h}^*$ является пороговым: $h_k^*(\boldsymbol{x}) = [\![\eta_k(\boldsymbol{x}) \geqslant t_k]\!]$, где $\eta_k(\boldsymbol{x}) = \mathbb{P}(y_k = 1|\boldsymbol{x})$, t_k — некоторый вектор порогов. Это позволяет применить двухступенчатый модульный подход для построения классификатора, на первом шаге которого обучается ранжирующая функция $\boldsymbol{\eta}\colon \mathbb{X} \to [0;\, 1]^n$, после чего на втором этапе определяется вектор оптимальных порогов $\boldsymbol{T} = (t_1, t_2, \ldots, t_n)$.

Методы, предложенные авторами в работах [Berger, Guda, 2020; Berger, Guda, 2021], посвящены второму этапу описанной выше процедуры. Зафиксировав ранжирующую функцию $\boldsymbol{\eta}$, они позволяют определить оптимальные положения порогов, максимизирующие выбранную метрику качества — меру F-macro.

3. Методы поиска оптимального вектора порогов

Для поиска оптимальных порогов в статьях [Berger, Guda, 2020; Berger, Guda, 2021] авторами были разработаны два алгоритма, эффективно работающие даже для задач экстремальной классификации. Они основаны на исследовании специально введенного отображения единичного квадрата на плоскости средних точности P и полноты R. Приведем краткое описание этих методов.

3.1. Основные определения

Определение 1. Будем называть $\boldsymbol{T}_0$ покоординатным максимумом, если $\forall k = 1 \ldots n$ функция $F\left(P\left(\boldsymbol{T}_0^{k,\tau}\right), R\left(\boldsymbol{T}_0^{k,\tau}\right)\right)$, зависящая от τ, достигает своего максимума, когда $\tau = t_k^0$.

Все максимумы любой функции являются покоординатными максимумами, обратное, вообще говоря, неверно. Этот факт лежит в основе следующего подхода: для решения поставленной задачи нужно найти все покоординатные максимумы функции F-macro($P(\boldsymbol{T})$, $R(\boldsymbol{T})$) и выбрать из них тот, которому соответствует наибольшее значение.

Определение 2. Определим отображение $\boldsymbol{W}$ пространства порогов как

$$\boldsymbol{W}(\boldsymbol{T}) = (w_1(\boldsymbol{T}), w_2(\boldsymbol{T}), \ldots, w_n(\boldsymbol{T})), \quad \text{где } w_k(\boldsymbol{T}) = \arg\max_{\tau} F\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right), \tag{6}$$

$$\boldsymbol{T}^{k,\tau} = (t_1, \ldots, t_{k-1}, \tau, t_{k+1}, \ldots, t_n). \tag{7}$$

Определение 3. Так как функции $w_k(\boldsymbol{T})$, вообще говоря, многозначны, то будем говорить, что вектор порогов $\boldsymbol{T}$ является неподвижной точкой отображения $\boldsymbol{W}$, если $\boldsymbol{T}$ принадлежит множеству значений $\boldsymbol{W}(\boldsymbol{T})$.

Неподвижные точки отображения $\boldsymbol{W}$, по определению, являются покоординатными максимумами функции $F(P(\boldsymbol{T}), R(\boldsymbol{T}))$.

Размерность пространства, в котором действует отображение $\boldsymbol{W}$, велика для многих задач, так как равна числу классов. Поэтому определение неподвижных точек в этих случаях может быть затруднительным. Чтобы упростить задачу, рассмотрим «двойник» отображения $\boldsymbol{W}$ — отображение $\boldsymbol{V}$, определенное на квадрате $[0; 1]^2$ так, что верно следующее равенство: $\boldsymbol{V} \circ (P, R) = (P, R) \circ \boldsymbol{W}$.

Определение 4. Пусть P, R — значения макроточности и полноты для некоторого вектора порогов $\boldsymbol{T}$. Определим $\boldsymbol{V}$ как $\boldsymbol{V}(P, R) = (P(\boldsymbol{W}(\boldsymbol{T})), R(\boldsymbol{W}(\boldsymbol{T})))$.

3.2. Метод линеаризации

Когда число классов n достаточно велико, мы можем заменить функцию F в формуле (6) ее разложением в ряд Тейлора в окрестности точки $(P, R) = (P(\boldsymbol{T}), R(\boldsymbol{T}))$:

$$\widehat{F}_{P,R}\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right) = F(P, R) + \frac{\partial F}{\partial P}(P, R)\left(P\left(\boldsymbol{T}^{k,\tau}\right) - P\right) + \frac{\partial F}{\partial R}(P, R)\left(R\left(\boldsymbol{T}^{k,\tau}\right) - R\right) \tag{8}$$

и определить соответствующие отображения $\widehat{\boldsymbol{W}}$ и $\widehat{\boldsymbol{V}}$ как

$$\begin{gathered}\widehat{\boldsymbol{W}} = (\widehat{w}_1(\boldsymbol{T}),\ \widehat{w}_2(\boldsymbol{T}),\ \ldots,\ \widehat{w}_n(\boldsymbol{T})),\\ \text{где } \widehat{w}_k(\boldsymbol{T}) = \arg\max_{\tau} \widehat{F}_{P(\boldsymbol{T}),\,R(\boldsymbol{T})}\left(P\left(\boldsymbol{T}^{k,\tau}\right),\ R\left(\boldsymbol{T}^{k,\tau}\right)\right),\\ \widehat{\boldsymbol{V}}(P,\,R) = (P(\widehat{\boldsymbol{W}}(\boldsymbol{T})),\ R(\widehat{\boldsymbol{W}}(\boldsymbol{T}))).\end{gathered} \tag{9}$$

В выражении (9), отбрасывая слагаемые, не зависящие от параметра τ, получим

$$\widehat{w}_k(\boldsymbol{T}) = \arg\max_{\tau} \frac{1}{n}\left[\frac{\partial F}{\partial P}(P(\boldsymbol{T}),\ R(\boldsymbol{T}))p_k(\tau) + \frac{\partial F}{\partial R}(P(\boldsymbol{T}),\ R(\boldsymbol{T}))r_k(\tau)\right]. \tag{10}$$

После вычисления частных производных

$$\frac{\partial F}{\partial P}(P,\,R) = \frac{2R^2}{(P+R)^2}, \quad \frac{\partial F}{\partial R}(P,\,R) = \frac{2P^2}{(P+R)^2}, \tag{11}$$

их подстановки в (10) и сокращения на множитель $\frac{2}{n(P+R)^2}$, не зависящий от τ, получим выражение

$$\widehat{w}_k(\boldsymbol{T}) = \arg\max_{\tau}\left(R^2(\boldsymbol{T})p_k(\tau) + P^2(\boldsymbol{T})r_k(\tau)\right). \tag{12}$$

Благодаря данному представлению отображение $\widehat{\boldsymbol{V}}(P,\,R)$ может быть естественным образом расширено на весь квадрат $[0;\,1]^2$. Его значения в точках области определения $\mathcal{D}(\boldsymbol{V})$ близки к значениям исходного отображения $\boldsymbol{V}$ (см. доказательство в [Berger, Guda, 2020]). Обозначим расширения на весь квадрат $(P,\,R) \in [0;\,1]^2$ отображений $\widehat{\boldsymbol{W}}$ и $\widehat{\boldsymbol{V}}$ соответствующими символами с волной

$$\begin{gathered}\widetilde{w}_k(P,\,R) = \arg\max_{\tau}\left(R^2 p_k(\tau) + P^2 r_k(\tau)\right),\\ \widetilde{\boldsymbol{V}}(P,\,R) = (P(\widetilde{\boldsymbol{W}}(P,\,R)),\ R(\widetilde{\boldsymbol{W}}(P,R))), \quad P,\,R \in [0;\,1].\end{gathered} \tag{13}$$

Важно отметить, что $\widetilde{\boldsymbol{V}}(P,\,R) = \widehat{\boldsymbol{V}}(P,\,R)$, когда $(P,\,R) \in \mathcal{D}(\boldsymbol{V})$, то есть существует вектор порогов $\boldsymbol{T}$ такой, что $P = P(\boldsymbol{T})$, $R = R(\boldsymbol{T})$.

Отображение $\widetilde{\boldsymbol{V}}$ является функцией полярного угла $\mathrm{Arg}(P,\,R)$, что позволяет нам сформулировать упрощенный алгоритм 1 поиска неподвижной точки. Он сводится к вычислению аргумента отображения $\mathrm{Arg}\,\widetilde{\boldsymbol{V}}(P,\,R)$ для набора точек с различными аргументами $\mathrm{Arg}(P,\,R)$, что позволяет быстро приближенно найти все неподвижные точки. Это является достаточным для многих практических приложений. Алгоритм 1′ (расширенный вариант алгоритма 1) возвращает оценку области, содержащей все неподвижные точки точного отображения $\boldsymbol{V}(P,\,R)$.

Алгоритм 1. Метод линеаризации

1: **Вход**: списки объектов для каждого класса c_k, отсортированные согласно некоторым ранжирующим функциям $\eta_k(\boldsymbol{x})$, $k = 1 \ldots n$; точные значения их меток $\boldsymbol{y}$.
2: Найти решение $(P^*,\,R^*)$ уравнения $\mathrm{Arg}(P,\,R) = \mathrm{Arg}\,\widetilde{\boldsymbol{V}}(P,\,R)$, имеющее максимальное значение метрики F-macro.
3: Найти оптимальный вектор порогов $T^* = \widetilde{\boldsymbol{W}}\,(P^*,\,R^*)$, где координаты $\widetilde{\boldsymbol{W}}$ определяются как $\widetilde{w}_k(P,\,R) = \arg\max_{\tau}\left(R^2 p_k(\tau) + P^2 r_k(\tau)\right)$.

Алгоритм 1′. Расширенный метод линеаризации

Вход: списки объектов для каждого класса c_k, отсортированные согласно некоторым ранжирующим функциям $\eta_k(\boldsymbol{x})$, $k = 1 \ldots n$; значения их меток $\boldsymbol{y}$.

Найти оптимальную точку

$$(P^*, R^*) = \underset{(P,R):\ (P,R)\in \mathrm{NeighbV}(P,R)}{\arg\max} F(P, R)$$

в области $\mathrm{NeighbV}(P, R)$, являющейся оценкой $\boldsymbol{V}(P, R)$ (см. определение ниже).

Найти оптимальный вектор порогов $\boldsymbol{T}^* = \widetilde{\boldsymbol{W}}(P^*, R^*)$, где координаты $\widetilde{\boldsymbol{W}}$ определены уравнением (13).

function NEIGHBV($(P, R) \in [0; 1]^2$)

Найти верхнюю границу ε для оценки модуля разности $|F - \widehat{F}|$ как $\varepsilon = \frac{2}{n^2}\frac{\left(P+R+\frac{2}{n}\right)^2}{\left(P+R-\frac{2}{n}\right)^3}$.

for $k = 1 \ldots n$ **do**

Для класса c_k найти множество $\varkappa_k$ порогов τ, для которых $\widehat{F}_{P(\boldsymbol{T}),R(\boldsymbol{T})}\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right)$ отличается от его максимума не более чем на 2ε:

$$\varkappa_k = \left\{\tau \mid \widehat{F}_{P(\boldsymbol{T}),R(\boldsymbol{T})}\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right) > \widehat{M}(P, R) - 2\varepsilon\right\}, \tag{14}$$

где $\widehat{M}(P, R) = \max\limits_{\tau} \widehat{F}_{P(\boldsymbol{T}),R(\boldsymbol{T})}\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right)$. Множество $\varkappa_k$ содержит точку максимума функции $F\left(P\left(\boldsymbol{T}^{k,\tau}\right), R\left(\boldsymbol{T}^{k,\tau}\right)\right)$ как функции от τ.

По формуле (13) найти оптимальный порог $t_k = \widetilde{w}_k(P, R)$ и соответствующее ему оптимальное значение $\widetilde{M}(P, R)$ линейной аппроксимации метрики F-macro.

Оценить интервалы $\left[p_k^{\min}; p_k^{\max}\right]$, $\left[r_k^{\min}; r_k^{\max}\right]$, содержащие максимальную точность и полноту класса c_k для точной функции F-macro:

$$p_k^{\min} = \min_{\tau\in\varkappa_k} p_k(\tau), \quad p_k^{\max} = \max_{\tau\in\varkappa_k} p_k(\tau); \quad r_k^{\min} = \min_{\tau\in\varkappa_k} r_k(\tau), \quad r_k^{\max} = \max_{\tau\in\varkappa_k} r_k(\tau). \tag{15}$$

end for

Пересчитать область $\left[P^{\min}; P^{\max}\right] \times \left[R^{\min}; R^{\max}\right]$, содержащую все значения точного отображения $\boldsymbol{V}(P, R)$:

$$P^{\min} = \frac{1}{n}\sum_{k=1}^{n} p_k^{\min}, \quad P^{\max} = \frac{1}{n}\sum_{k=1}^{n} p_k^{\max}; \quad R^{\min} = \frac{1}{n}\sum_{k=1}^{n} r_k^{\min}, \quad R^{\max} = \frac{1}{n}\sum_{k=1}^{n} r_k^{\max}. \tag{16}$$

return $\left[P^{\min}; P^{\max}\right] \times \left[R^{\min}; R^{\max}\right]$.

end function

3.3. Метод анализа области определения V

В основе этого метода лежит анализ области определения отображения $\boldsymbol{V}$:

$$\mathcal{D}(\boldsymbol{V}) = \left\{(P, R) \mid \exists \boldsymbol{T}\ P = P(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{n} p_k(t_k),\ R = R(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{n} r_k(t_k)\right\}. \tag{17}$$

Она имеет сложную дискретную структуру. Координаты точек $(P, R) \in \mathcal{D}(\boldsymbol{V})$ являются средними значений точности p_k или полноты r_k для некоторых порогов t_k в классах c_k, $k = 1, \ldots, n$. Если

построить хорошее приближение к области $\mathcal{D}(\boldsymbol{V})$, то максимизация меры $F(P, R)$ на ней позволит найти оптимальные точность P и полноту R, по которым можно определить вектор порогов.

Проанализировать все точки области определения не представляется возможным: даже в случае 1000 объектов и 10 классов, что является примером достаточно небольшой задачи, размер множества $\mathcal{D}(\boldsymbol{V})$ может достигать 1000^{10}. Поэтому рассмотрим выпуклую оболочку $H = \operatorname{hull}\mathcal{D}(\boldsymbol{V})$. Так как функция $F(P, R)$, определенная на $[0; 1]^2$, не имеет внутренних максимумов в $[0; 1]^2$, то максимум на множестве H достигается на его границе, что позволяет получить оценку сверху значения максимума $F(P, R)$ на $\mathcal{D}(\boldsymbol{V})$. А вершины выпуклой оболочки H дают оценку максимума снизу.

Теорема 1 позволяет сократить число рассматриваемых при оптимизации точек, предоставляет верхнюю и нижнюю оценки оптимума и является обоснованием второго алгоритма поиска вектора оптимальных порогов.

Теорема 1. *Пусть* $\Gamma \subset \mathcal{D}(\boldsymbol{V})$ *есть множество вершин выпуклой оболочки* H. *Тогда*

$$\max_{\Gamma} F \leqslant \max_{\mathcal{D}(\boldsymbol{V})} F \leqslant \max_{H} F. \tag{18}$$

На рис. 1 видно, насколько сильно увеличивается плотность точек, принадлежащих области определения. Построение выпуклой оболочки области, имеющей такую сложную дискретную структуру, представляет собой отдельную задачу. Быстрое переполнение памяти для датасетов экстремальной многозначной классификации не дает решать эту задачу «в лоб». Последовательное построение выпуклой оболочки, когда в цикле для каждого класса сначала строится выпуклая оболочка его значений точности и полноты $h_k = \operatorname{hull}\left(\{p_k(\tau),\ r_k(\tau)\}_{\tau=1}^{m}\right)$, которая по рекурсивной формуле

$$H_{k+1} = \operatorname{hull}\left(H_k + \operatorname{hull}\left(\{p_k(\tau),\ r_k(\tau)\}_{\tau=1}^{m}\right)\right)$$

затем прибавляется к накапливаемой оболочке всех рассмотренных классов H_k:

$$H_k = \operatorname{hull}\left\{(P, R) \mid \exists \boldsymbol{T}\ P = P(\boldsymbol{T}) = \frac{1}{k}\sum_{i=1}^{k} p_i(t_i),\ R = R(\boldsymbol{T}) = \frac{1}{k}\sum_{i=1}^{k} r_i(t_i)\right\}, \quad k = 1, 2, \ldots, n,$$

не является достаточно вычислительно эффективным при наличии вычислительной сложности $O\left(nm\sqrt{n \ln m}\right)$ (см. [Berger, Guda, 2021]).

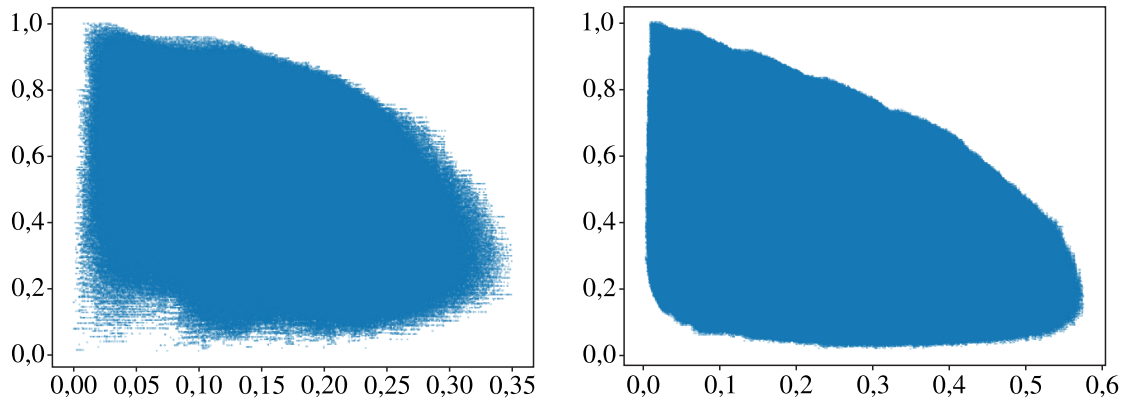


Рис. 1. Область определения $\mathcal{D}(\boldsymbol{V})$ для 5 (а) и 10 (б) классов набора данных ESP Game

Чтобы добиться быстрой работы алгоритма даже для задач экстремальной классификации, рассмотрим n_α полярных углов ϕ (направлений) и соответствующих им крайних точек $(P_e(\phi), R_e(\phi))$ области определения отображения $\boldsymbol{V}$:

$$(P_e(\phi),\, R_e(\phi)) = \underset{(P,R)\in\mathcal{D}(\boldsymbol{V})}{\arg\max}\, (P\cos\phi + R\sin\phi). \tag{19}$$

Использование аппроксимации H^α выпуклой оболочки H области $\mathcal{D}(\boldsymbol{V})$ —

$$H^\alpha = \left\{(P,\, R) \mid P\cos\phi + R\sin\phi \leqslant P_e(\phi)\cos\phi + R_e(\phi)\sin\phi,\ \forall\phi = \frac{2\pi\ell}{n_\alpha},\ \ell = 1\ldots n_\alpha\right\}, \tag{20}$$

— позволяет существенно сократить вычисления. Пусть Γ^α — множество точек $(P_e(\phi), R_e(\phi))$, $\phi = \frac{2\pi\ell}{n_\alpha}$, $\ell = 1\ldots n_\alpha$, тогда теорема 1 будет верна для приближенных множеств H^α и Γ^α.

В результате в работе [Berger, Guda, 2021] был разработан следующий алгоритм быстрого построения приближения выпуклой оболочки H^α — алгоритм 2, — имеющий сложность $O(n_\alpha nm)$ и позволяющий осуществлять вычисления на обычном персональном компьютере.

Алгоритм 2. Быстрое построение hull $\mathcal{D}(\boldsymbol{V})$

1: **Вход**: списки объектов для каждого класса c_k, отсортированные согласно некоторым ранжирующим функциям $\eta_k(\boldsymbol{x})$, $k = 1, 2, \ldots, n$; значения их меток $\boldsymbol{y}$, число полярных углов n_α.
2: Инициализировать аппроксимацию множества вершин выпуклой оболочки случайным образом, например нулями $\Gamma^\alpha = \{(0, 0)\}_{\ell=1}^{n_\alpha}$.
3: **for** $k = 1, \ldots, n$ **do**
4: $\quad h_k = \text{hull}\left(\{p_k(\tau),\, r_k(\tau)\}_{\tau=1}^{m}\right)$.
5: $\quad$ Перебирать точки множества h_k против часовой стрелки:
6: $\quad$ **for** $(p_i,\, r_i)$ in h_k **do**
7: $\quad\quad$ Найти полярный угол ϕ_{prev} нормали к отрезку, соединяющему точки $(p_{i-1},\, r_{i-1})$ и $(p_i,\, r_i)$, и полярный угол ϕ_{next} нормали к отрезку между точками $(p_i,\, r_i)$ и $(p_{i+1},\, r_{i+1})$.
8: $\quad\quad$ Пересчитать вершины $(P_e(\phi), R_e(\phi)) \in \Gamma_\alpha$ для всех углов $\phi = \frac{2\pi\ell}{n_\alpha} \in [\phi_{\text{prev}},\, \phi_{\text{next}}]$:

$$P_e(\phi) = \frac{P_e(\phi)\cdot(k-1) + p_i}{k}, \quad R_e(\phi) = \frac{R_e(\phi)\cdot(k-1) + r_i}{k}. \tag{21}$$

9: $\quad$ **end for**
10: **end for**
11: Найти точку $(P^*,\, R^*) \in \Gamma^\alpha$, в которой F-macro достигает максимума, — нижняя граница максимума F-macro на $\mathcal{D}(\boldsymbol{V})$.
12: Найти точку $\left(P_H^*,\, R_H^*\right) \in H^\alpha$, в которой F-macro достигает максимума, — верхняя граница максимума F-macro на $\mathcal{D}(\boldsymbol{V})$.
13: Вычислить оптимальный вектор порогов $\boldsymbol{T}^* = \boldsymbol{W}(P^*,\, R^*)$, соответствующий $(P^*,\, R^*)$.

4. Исследование свойств алгоритмов поиска оптимальных векторов порогов

4.1. Наборы данных

Для исследования свойств разработанных алгоритмов использовалось три набора данных из различных областей: WikiLSHTC-325K [Partalas et al., 2015] (текст), ESP Game [Von Ahn, Dabbish, 2004] (изображения) и искусственно сгенерированный набор данных Example, использованный для исследования некоторых свойств алгоритма 1′ в статье [Berger, Guda, 2020].

Набор данных ESP Game получен путем онлайн-игры — разметки ESP, в которой два игрока зарабатывают очки, пытаясь описать изображение с помощью того же ключевого слова, не общаясь между собой. В эксперименте используется 20 770 изображений, описанных $n = 268$ ключевыми словами, разбитых на тренировочное и тестовое множества, содержащие 18 689 и 2 081 изображений соответственно. Для этого набора данных извлечение признаков было произведено с помощью нейронной сети на основе модели Xception [Chollet, 2017], обученной на тренировочном множестве и использующей в качестве стартовой точки веса, полученные путем обучения на наборе изображений ImageNet [Deng et al., 2009]. Тренировка нейронной сети осуществлялась с помощью библиотеки автоматического дифференцирования и обучения нейронных сетей Pytorch [Paszke et al., 2019].

Набор данных WikiLSHTC-325K содержит 2 365 435 документов, отнесенных к $n = 325\,056$ категориям. Он служит примером задачи экстремальной многозначной классификации [Bhatia et al., 2016]. Тренировочное множество содержит 1 778 351 документ, тестовое множество — 587 084 документа. В качестве ранжирующего алгоритма для этого набора данных был использован метод отбора подобных по коэффициенту склонности PfastreXML [Jain, Prabhu, Varma, 2016]. По умолчанию он предсказывает 5 категорий каждому объекту, но в ходе экспериментов было выявлено, что увеличение этого числа до 1000 предсказываемых категорий приводит к балансу между качеством получившегося ранжирования и скоростью поиска оптимальных порогов.

Искусственно сгенерированный набор данных Example представляет собой набор из $m = 7$ абстрактных объектов. Ранжирующая функция упорядочивает объекты в каждом классе *одинаковым образом* так, что получившийся упорядоченный список меток объектов имеет следующий вид: 1, 0, 0, 0, 1, 1, 0, где 1 — верно классифицированный объект, 0 — неверно. Такой набор данных был сформирован с целью демонстрации интересных особенностей работы предложенных алгоритмов в случае, когда предсказания ранжирующей функции совпадают во множестве классов и для каждого из них существует несколько оптимальных позиций порогов с точки зрения F-macro.

4.2. Особенности программной реализации

Для реализации программного комплекса был выбран высокоуровневый язык программирования Python [Van Rossum, 1995] с использованием библиотеки высокопроизводительных вычислений Numba [Lam, Pitrou, Seibert, 2015], генерирующей оптимизированный машинный код с использованием компилятора LLVM [Lattner, Adve, 2004], что позволяет программам сравниться по скорости с программами, написанными на компилируемых языках.

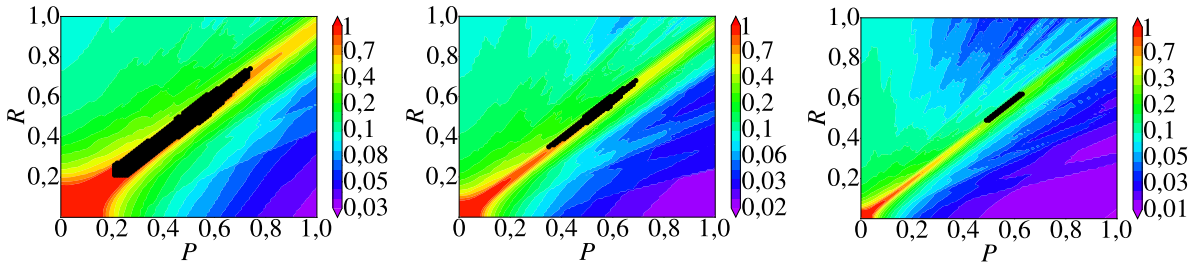
Расширенный метод линеаризации, описанный в алгоритме 1′, представлен скриптом `run_V_PR_estimation.py`. Метод анализа области определения $\mathcal{D}(\boldsymbol{V})$, описанный в алгоритме 2, реализован скриптом `run_DV_estimation.py`. Предполагается, что функция, ранжирующая объекты, в каждом классе задана. Программы нуждаются лишь в информации об истинных метках, упорядоченных в каждом классе объектов: метка 1 означает верно классифицированный объект, 0 — неверно классифицированный. Обе программы принимают на вход файл, где каждая строка соответствует классу, 0 и 1 соответствуют истинным меткам проранжированных для данного класса объектов. Метод анализа области определения требует также указать число полярных направлений n_α, по которым будет строиться приближенная выпуклая оболочка искомой области. Каждая из программ может работать для любого $\beta > 0$ из метрики F_β-macro. Результатом работы расширенного метода линеаризации является область, содержащая все неподвижные точки точного отображения $\boldsymbol{V}(P, R)$. В результате работы `run_DV_estimation.py` строится выпуклая оболочка с n_α вершинами. Вершина с максимальным значением метрики F-macro является нижней границей оценки максимума F-macro.

Код программ и подготовленные входные файлы для наборов данных WikiLSHTC-325K, ESP Game и Example (папка `data`) выложены в открытый доступ в составе Github-репозитория `f_macro_optimization` [Berger, Guda, 2022].

Все вычислительные эксперименты были проведены на персональном компьютере, оснащенном оперативной памятью DDR4 объемом 32 Гб и процессором Intel(R) Core(TM) i7-8700 CPU @ 3,20 GHz, имеющим 6 физических ядер.

4.3. Погрешность алгоритмов в зависимости от числа классов

Погрешности алгоритмов будем измерять в терминах макровеличин: P, R, $F(P, R)$, а также аргумента и полярного радиуса точки (P, R). Алгоритм 1 находит оптимальное решение для линеаризованного варианта метрики F-macro. Используя расширенный алгоритм 1′, можно оценить область, которой принадлежат значение точного отображения $\boldsymbol{V}(P, R)$ и все его неподвижные точки.



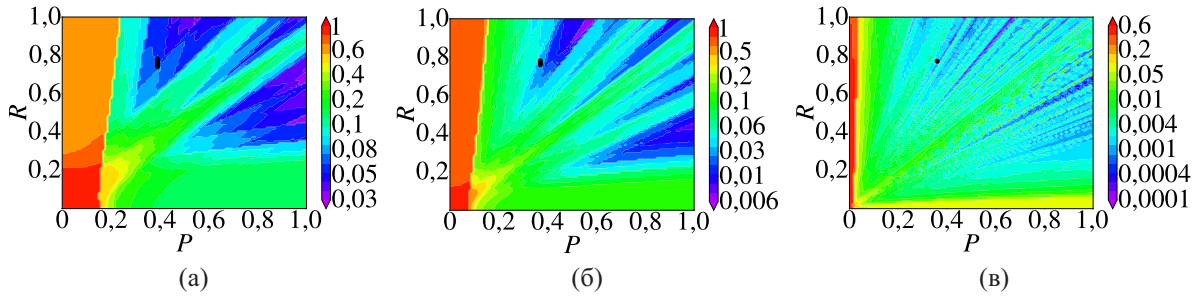
(а) (б) (в)

Рис. 2. Оценка абсолютной ошибки $|\boldsymbol{V}(P, R) - \widetilde{\boldsymbol{V}}(P, R)|$, допущенной алгоритмом 1′ для набора данных ESP Game и числа классов $n = 50$ (а), 100 (б), 200 (в). Цветом отмечены значения $\sqrt{(P^{\max} - P^{\min})^2 + (R^{\max} - R^{\min})^2}$. Черная область содержит все неподвижные точки отображения $\boldsymbol{V}(P, R)$: $P \in \left[P^{\min}; P^{\max}\right]$, $R \in \left[R^{\min}; R^{\max}\right]$

Рис. 2 иллюстрирует отличие точного $\boldsymbol{V}(P, R)$ и приближенного $\widetilde{\boldsymbol{V}}(P, R)$ отображений: $|\boldsymbol{V}(P, R) - \widetilde{\boldsymbol{V}}(P, R)|$ для $n = 50$ (а), 100 (б), 200 (в) классов набора данных ESP Game. Черным цветом отмечена область, точки которой потенциально могут быть неподвижными для отображения $\boldsymbol{V}(P, R)$.

В то время как для 50 классов область, содержащая все неподвижные точки, достаточно велика, с ростом числа классов она значительно сужается до небольшого вытянутого вдоль прямой $R = P$ участка. Вытянутость областей, соответствующих наибольшим по модулю значениям отклонения $\widetilde{\boldsymbol{V}}(P, R)$ от $\boldsymbol{V}(P, R)$ (отмеченных красным и оранжевым), вдоль прямой $R = P$ объясняется значительным процентным содержанием классов с несколькими оптимальными положениями порогов для равных средней точности P и полноты R.

Для набора данных WikiLSHTC-325K (см. рис. 3) отмеченная черным цветом область, которой принадлежат все неподвижные точки отображения $\boldsymbol{V}(P, R)$, расположена в районе с относительно маленькой погрешностью $|\boldsymbol{V}(P, R) - \widetilde{\boldsymbol{V}}(P, R)|$. Из-за этого она имеет меньший размер по сравнению с датасетом ESP Game, еще более сокращающийся по мере увеличения числа классов. Для $n = 1000$ расстояние между $\boldsymbol{V}(P, R)$ и $\widetilde{\boldsymbol{V}}(P, R)$ в среднем приближается к значению 0,01. Красно-оранжевая область здесь также возникает из-за большого процента классов, для которых существует несколько оптимальных положений порогов для значений (P, R) из этой области. Вариация в выборе порога приводит к большим возможным отличиям $\boldsymbol{V}(P, R)$ и $\widetilde{\boldsymbol{V}}(P, R)$.



(а) (б) (в)

Рис. 3. Оценка абсолютной ошибки $|\boldsymbol{V}(P, R) - \widetilde{\boldsymbol{V}}(P, R)|$, допущенной алгоритмом 1′ для набора данных WikiLSHTC-325K и числа классов $n = 50$ (а), 100 (б), 1000 (в). Цветом обозначены значения $\sqrt{(P^{\max} - P^{\min})^2 + (R^{\max} - R^{\min})^2}$. Черная область содержит все неподвижные точки отображения $\boldsymbol{V}(P, R)$: $P \in \left[P^{\min};\, P^{\max}\right]$, $R \in \left[R^{\min};\, R^{\max}\right]$

Представленные результаты позволяют говорить о целесообразности применения алгоритма 1 в первую очередь для задач, где число классов достаточно велико, то есть для задач экстремальной многозначной классификации.

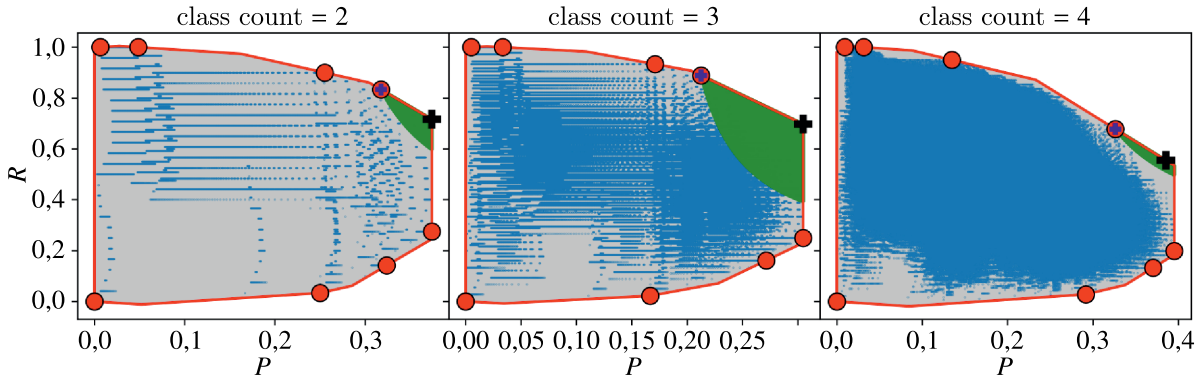


Рис. 4. Область определения $\mathcal{D}(\boldsymbol{V})$ и найденные оценки оптимального значения F-macro (синий и черный кресты — оценки снизу и сверху соответственно) для 2, 3 и 4 первых классов ESP Game и фиксированного числа полярных углов $n_\alpha = 14$. Синие точки — истинная область определения $\boldsymbol{V}$; красный цвет — выпуклая оболочка H_α, построенная алгоритмом 2; зеленая область — найденная область расположения максимума F-macro (точки (P, R) многогранника H_α, для которых значения F-macro больше оценки оптимума снизу)

Алгоритм 2 подходит к задаче поиска оптимальных порогов с другой стороны. Его результатами являются нижняя и верхняя оценки оптимального значения меры F-macro и набор порогов, соответствующий нижней оценке. Алгоритм 2 не использует линеаризованный вариант меры F-macro, а находит аппроксимацию выпуклой оболочки области определения точного отображения $\boldsymbol{V}(P, R)$. Рис. 4 иллюстрирует результат работы алгоритма 2 для разных значений числа классов. Эта визуализация требует поиска всех точек области определения, что становится вычислительно сложной задачей уже для малого числа классов, поэтому выбранные для построения рисунков количества классов невелики: $n = 2, 3, 4$. С другой стороны, видно, что уже для числа классов $n = 4$ область, содержащая точки, для которых значение F-macro больше оценки оптимума снизу, достаточно мала, и видна тенденция к ее постепенному, хотя и не монотонному уменьшению. Отсюда не следует вывод о том, что алгоритм 2 применим только к задачам с маленьким числом классов. Как показывает оценка сложности алгоритма 2 (см. конец подпараграфа 3.3), он применим и к задачам с большим числом классов. Проблема только

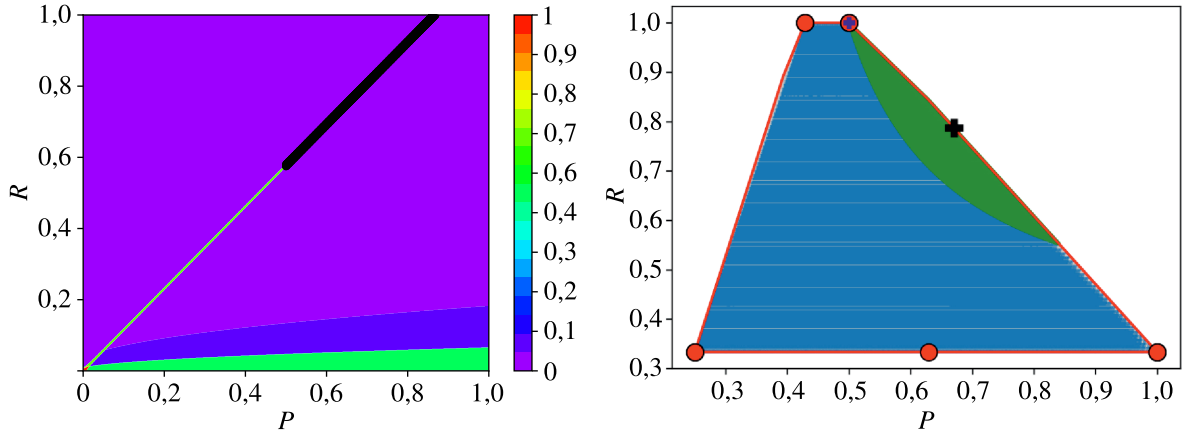


Рис. 5. (а) оценка абсолютной ошибки $|V(P, R) - \widetilde{V}(P, R)|$, допущенной алгоритмом 1′ для набора данных Example и числа классов $n = 80$; (б) область определения $\mathcal{D}(\boldsymbol{V})$ и найденные оценки оптимального значение F-macro (синий и черный кресты — оценки снизу и сверху соответственно) для числа классов $n = 80$ и числа полярных углов $n_\alpha = 100$

в поиске всех точек области определения и отображения их на графике, так как их количество экспоненциально растет с увеличением числа классов.

Алгоритм 2, в отличие от алгоритма 1′, с успехом применяется для определения оптимального числа порогов как в задачах с небольшим числом классов, так и для экстремальной многозначной классификации. Но всегда ли использование алгоритма 2 предпочтительнее? Набор данных Example становится проблемой для алгоритма 2, что демонстрируется на рис. 5: нижняя оценка меры F-macro сильно отличается от верхней даже для достаточно большого количества классов $n = 80$, что объясняется наличием длинных прямых участков границы области определения. Эта проблема не упраздняется с увеличением числа классов. В подобных случаях алгоритм 1′ гораздо лучше определяет полярный угол оптимальной точки, как видно из рис. 5, *а*.

Таблица 1. Погрешности работы методов линеаризации и анализа области определения $\mathcal{D}(\boldsymbol{V})$ для наборов данных ESP Game, WikiLSHTC и Example в зависимости от числа классов n (число полярных углов фиксировано: $n_\alpha = 100$): разброс значений точности (dP), полноты (dR), меры F-macro (dF-macro), полярного радиуса (dRad) и полярного угла (dArg) для возвращаемой ими области нахождения оптимума F-macro

Набор данных	Метод	n	dP	dR	dF-macro	dRad	dArg
ESP Game	линеаризации	50	0,53	0,53	0,53	0,75	0,23
		100	0,34	0,33	0,335	0,47	0,045
		200	0.14	0.13	0.135	0.19	0.01
	$\mathcal{D}(\boldsymbol{V})$	50	0,00	0,00	0,00	0,00	0,00
		100	0,00	0,00	0,00	0,00	0,00
		200	0,14	0,14	0,01	0,02	0,26
WikiLSHTC	линеаризации	50	0,0	0,04	0,009	0,04	0,02
		100	0,00	0,02	0,004	0,02	0,01
		1000	0,00	0,00	0,00	0,00	0,00
	$\mathcal{D}(\boldsymbol{V})$	50	0,00	0,00	0,00	0,00	0,00
		100	0,00	0,01	0,00	0,01	0,01
		1000	0,00	0,01	0,00	0,01	0,01
Example	линеаризации	1000	0,36	0,41	0,38	0,55	0,005
	$\mathcal{D}(\boldsymbol{V})$	1000	0,34	0,44	0,06	0,17	0,52

Сравнить работу алгоритмов 1′ и 2 также позволяет сводная таблица 1, в которой приведены ошибки методов линеаризации и анализа области определения $\mathcal{D}(\boldsymbol{V})$ для наборов данных ESP Game, WikiLSHTC и Example и разного числа классов n. Эти ошибки позволяют оценить возвращаемую алгоритмами область, содержащую оптимум меры F-macro, по нескольким параметрам: разница между максимальными и минимальными точностью (dP), полнотой (dR), значением F-macro (dF-macro), полярным радиусом (dRad) и полярным углом (dArg).

Численные значения, приведенные в таблице 1, подтверждают выводы о том, что в задачах экстремальной многозначной классификации, где число классов и объектов велико (WikiLSHTC), методы линеаризации и анализа области определения имеют одинаково небольшие погрешности, в то время как в задачах небольшой размерности (ESP Game) именно метод анализа области определения приводит к меньшим значениям погрешностей по всем измеренным параметрам.

Применение алгоритмов к набору данных Example демонстрирует, что метод анализа области определения испытывает трудности в случае, когда область определения имеет протяженные прямые границы: предполагаемая область расположения оптимума остается достаточно большой и не уменьшается с ростом числа классов n. Метод линеаризации также возвращает достаточно большую погрешность для набора данных Example по всем параметрам, кроме полярного угла.

4.4. Погрешность алгоритмов в зависимости от β

На рисунке 6 представлена зависимость погрешности определения оптимального значения меры F_β-macro от значения β. Когда параметр β лежит на интервале (0, 1), более предпочтительной является оптимизация точности, в противном случае ($\beta > 1$) — полноты.

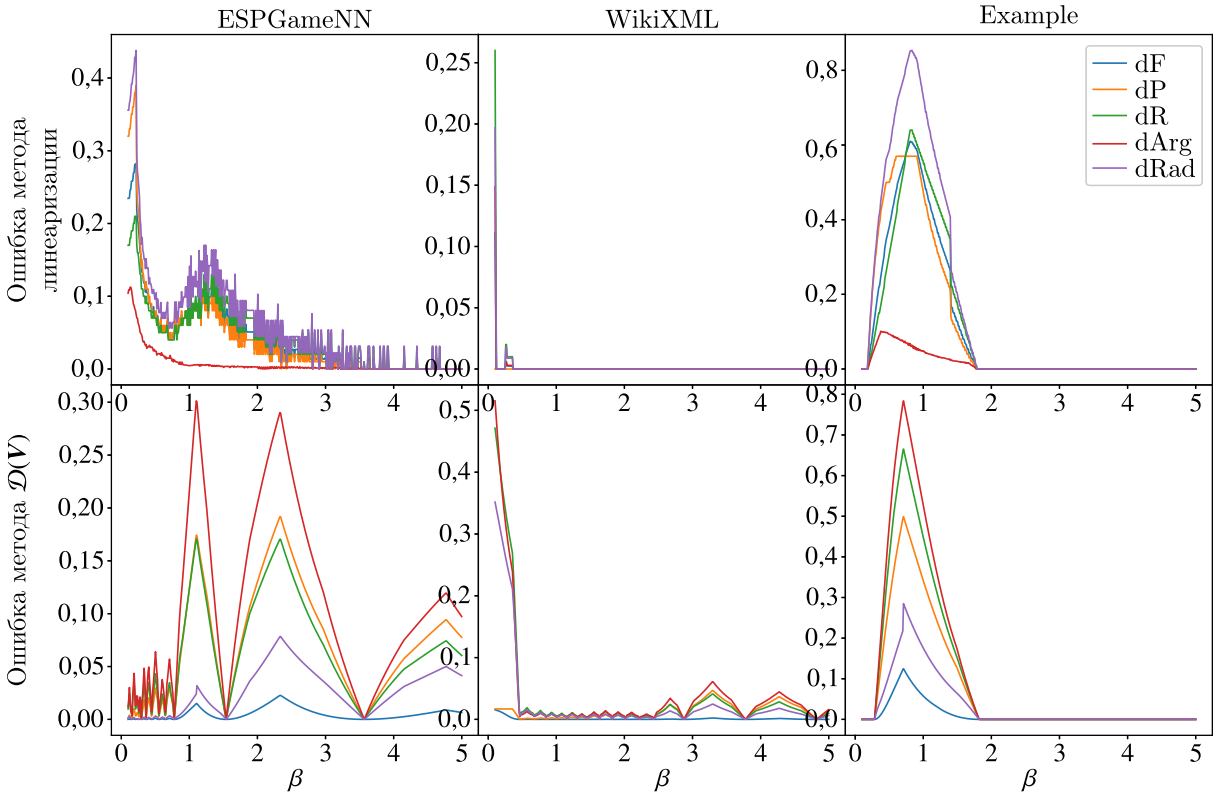


Рис. 6. Погрешности определения максимума F_β-macro в зависимости от β для наборов данных ESPGame, WikiLSHTC и Example с использованием алгоритмов 1′ и 2

Можно заметить, что два метода, линеаризации и анализа $\mathcal{D}(\boldsymbol{V})$, совершенно по-разному аппроксимируют область, содержащую неподвижную точку. Метод линеаризации точнее всего

определяет аргумент оптимальной точки (P, R) и хуже всего — полярный радиус (см. ошибки dF и dRad на рис. 6). Метод анализа области определения $\boldsymbol{V}$ работает противоположным образом: лучше всего определяет радиус и хуже всего — полярный угол. Таким образом, применять оба алгоритма следует в связке: метод линеаризации — для определения аргумента оптимальной точки, а метод анализа $\mathcal{D}(\boldsymbol{V})$ — для поиска полярного радиуса.

4.5. Погрешность алгоритма 2 в зависимости от числа полярных углов

Проанализируем, как изменяется точность работы алгоритма 2 от числа полярных углов для малого числа классов в наборе данных. Рис. 7 иллюстрирует, что уже для 5 классов и всего 20 направлений найденная алгоритмом 2 точка максимума близка к абсолютному оптимуму. Постепенное увеличение количества полярных углов n_α до 200 позволяет уменьшить значение погрешности $\mathrm{Err} = \sqrt{(P^{\max} - P^{\min})^2 + (R^{\max} - R^{\min})^2}$ до 0,04.

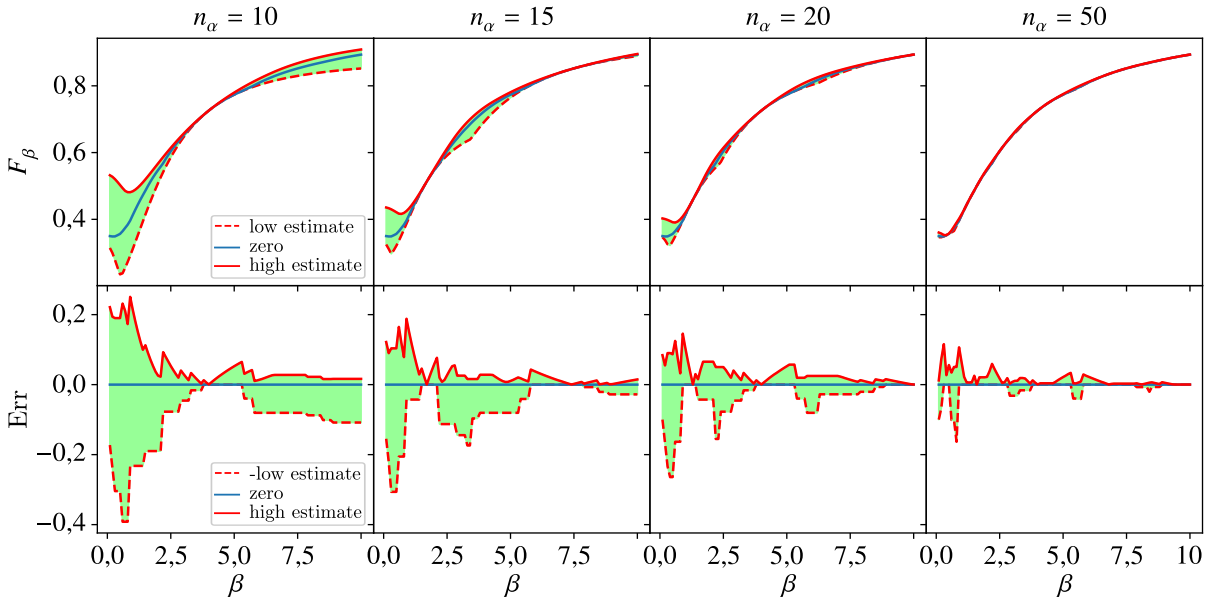


Рис. 7. Зависимость оптимального значения F_β-macro и значения погрешности $|V(P, R) - \widetilde{V}(P, R)|$ от значения β для разных значений количества полярных углов для набора данных ESPGame. Число классов фиксировано и равно 5

Однако уменьшение погрешности при увеличении количества полярных углов не является общей закономерностью. Для датасета Example из-за наличия протяженных прямых границ области определения (см. рис. 5) ошибка не уменьшается с ростом числа полярных углов n_α.

5. Заключение

Построение моделей многозначной классификации является сложной, хотя и востребованной на практике задачей. F-мера является популярной оценкой качества для задачи многозначной классификации. Популярный в области категоризации документов вариант F-меры не допускает независимой оптимизации положений порогов в каждом классе, что мотивировало авторов на разработку двух эффективных алгоритмов оптимизации: метода линеаризации и метода анализа области определения особым образом введенного отображения квадрата $[0; 1]^2$. Алгоритмы применимы в том числе и к задачам экстремальной классификации. В работе был проведен сравнительный анализ двух алгоритмов поиска оптимальных порогов для моделей многозначной

классификации и изучались границы применимости путем проведения вычислительных экспериментов на данных различной природы. Дополнительной сложностью является то, что в задачах экстремальной многозначной классификации определение оптимальных порогов происходит в пространстве, размерность которого может превышать сотни тысяч, что осложняет применение классических методов оптимизации, которые к тому же не дают теоретических оценок на оптимальность полученного решения для фиксированной ранжирующей функции (в отличие от предложенных авторами алгоритмов).

Погрешности исследуемых алгоритмов изучены применительно к наборам данных разного размера и из различных предметных областей: WikiLSHTC, ESP Game и искусственно сгенерированному набору данных Example. Обнаружено, что метод анализа $\mathcal{D}(\boldsymbol{V})$ для небольшого количества классов приводит в среднем к меньшей погрешности в определении оптимальных значений средней точности P и полноты R, чем метод линеаризации. Для задач экстремальной классификации точность обоих методов приблизительно одинакова.

Выявлена также особенность работы обоих алгоритмов для задач с областью $\mathcal{D}(\boldsymbol{V})$, содержащей протяженные линейные участки границ. В случае когда оптимальная точка расположена в окрестности этих участков, погрешности обоих методов не уменьшаются с увеличением количества классов. При этом метод линеаризации достаточно точно определяет аргумент оптимальной точки, а метод анализа $\mathcal{D}(\boldsymbol{V})$ — полярный радиус.

Вычислительные эксперименты демонстрируют, что разработанные алгоритмы дают небольшие погрешности для наборов данных из реальной жизни, а значит, эффективно решают задачу поиска оптимальных порогов для моделей многозначной классификации.

Список литературы (References)

Данилов Г. В., Жуков В. В., Куликов А. С., Макашова Е. С., Митин Н. А., Орлов Ю. Н. Сравнительный анализ статистических методов классификации научных публикаций в области медицины // Компьютерные исследования и моделирование. — 2020. — Т. 12, № 4. — С. 921–933.
Danilov G. V., Zhukov V. V., Kulikov A. S., Makashova E. S., Mitin N. A., Orlov Yu. N. Sravnitelnyj analiz statisticheskih metodov klassifikacii nauchnyh publikacij v oblasti [Comparative analysis of statistical methods of scientific publications classification in medicine] // Computer research and modelling. — 2020. — Vol. 12, No. 4. — P. 921–933 (in Russian).

Орлова Е. В. Оценка кредитного риска на основе методов многомерного анализа // Компьютерные исследования и моделирование. — 2013. — Т. 5, № 5. — С. 893–901.
Orlova E. V. Ocenka kreditnogo riska na osnove metodov mnogomernogo analiza [Credit risk assessment on the basis of multidimensional analysis] // Computer research and modelling. — 2013. — Vol. 5, No. 5. — P. 893–901 (in Russian).

Сабиров А. И., Катасёв А. С., Дагаева М. В. Нейросетевая модель распознавания знаков дорожного движения в интеллектуальных транспортных системах // Компьютерные исследования и моделирование. — 2021.— Т. 13, № 2. — С. 429–435.
Sabirov A. I., Katasev A. S., Dagaeva M. V. Nejrosetevaya model' raspoznavaniya znakov dorozhnogo dvizheniya v intellektual'nyh transportnyh sistemah [A neural network model for traffic signs recognition in intelligent transport systems] // Computer research and modelling. — 2021. — Vol. 13, No. 2. — P. 429–435 (in Russian).

Berger A. I., Guda S. A. F-macro optimization [Electronic resource]: https://github.com/melaanya/f_macro_optimization (дата обращения: 08.06.2022).

Berger A. I., Guda S. A. Multi-label classification based on domain analysis in fixed point method // 2021 28th Conference of Open Innovations Association (FRUCT). — 2021. — P. 1–7.

Berger A. I., Guda S. A. Threshold optimization for F measure of macro-averaged precision and recall // Pattern Recognition. — 2020. — Vol. 102. — P. 107250.

Bhatia K., Dahiya K., Jain H., Kar P., Mittal A., Prabhu Y., Varma M. The extreme classification repository: Multi-label datasets and code // 2016. — http://manikvarma.org/downloads/XC/XMLRepository.html

Chollet F. Xception: Deep learning with depthwise separable convolutions // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — P. 1251–1258.

Cornolti M., Ferragina P., Ciaramita M. A framework for benchmarking entity-annotation systems // Joint Proceedings of the 22nd international conference on World Wide Web. — 2013. — P. 249–260.

Decubber S., Mortier T., Dembczyński K., Waegeman W. Deep f-measure maximization in multi-label classification: A comparative study // Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. — 2018. — P. 290–305.

Dembczynski K., Jachnik A. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization // International conference on machine learning, PMLR. — 2013. — P. 1130–1138.

Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L. Imagenet: A large-scale hierarchical image database // 2009 IEEE conference on computer vision and pattern recognition. — 2009. — P. 248–255.

Fan R. E., Lin C. J. A study on threshold selection for multi-label classification // Department of Computer Science, National Taiwan University. — 2007. — P. 1–23.

Jain H., Prabhu Y., Varma M. Extreme multi-label loss functions for recommendation, tagging, ranking and other missing label applications // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — 2016. — P. 935–944.

Koyejo O., Natarajan N., Ravikumar P., Dhillon I. S. Consistent Multilabel Classification // NIPS. — 2015. — Vol. 29. — P. 3321–3329.

Lagutina K. L. et al. Sentiment classification of russian texts using automatically generated thesaurus // 2018 23rd Conference of Open Innovations Association (FRUCT). — 2018. — P. 217–222.

Lam S. K., Pitrou A., Seibert S. Numba: A llvm-based python jit compiler // Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. — 2015. — P. 1–6.

Lattner C., Adve V. LLVM: A compilation framework for lifelong program analysis & transformation // International Symposium on Code Generation and Optimization. — 2004. — P. 75–86.

Lipton Z. C., Elkan C., Naryanaswamy B. Optimal thresholding of classifiers to maximize F1 measure // Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer. — 2014. — P. 225–239.

Luo L., Li L. Defining and evaluating classification algorithm for high-dimensional data based on latent topics // PloS one. — 2014. — Vol. 9, No. 1. — P. 82119.

Partalas I. Kosmopoulos A., Baskiotis N., Artieres T., Paliouras G., Gaussier E. Lshtc: A benchmark for large-scale text classification // arXiv preprint arXiv:1503.08581. — 2015.

Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G. Pytorch: An imperative style, high-performance deep learning library // Advances in neural information processing systems. — 2019. — Vol. 32.

Pillai I., Fumera G., Roli F. Designing multi-label classifiers that maximize F measures: State of the art // Pattern Recognition. — 2017. — Vol. 61. — P. 394–404.

Pillai I., Fumera G., Roli F. Threshold optimisation for multi-label classifiers // Pattern Recognition. — 2013. — Vol. 46. — P. 2055–2065.

Van Rijsbergen C. J. Information Retrieval. — 2nd ed. — Butterworths, 1979. — 208 p.

Tran D., Mac H., Tong V., Tran H. A., Nguyen L. G. A LSTM based framework for handling multiclass imbalance in DGA botnet detection // Neurocomputing. — 2018. — Vol. 275. — P. 2401–2413.

Van Rossum G. Python reference manual // Department of Computer Science [CS]. — 1995. — No. R9525.

Von Ahn L., Dabbish L. Labeling images with a computer game // Proceedings of the SIGCHI conference on Human factors in computing systems. — 2004. — P. 319–326.

Yang Y. A study of thresholding strategies for text categorization // Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. — 2001. — P. 137–145.