Multi-label Classification Based on Domain Analysis in Fixed Point Method

Anna Berger, Sergey Guda
Southern Federal University
Rostov-on-Don, Russia
anna.ig.berger, gudasergey@gmail.com

***Abstract*—Nowadays a multi-label classification problem arises in different areas for which the significant amount of data has been gained. This problem can be viewed as the one comprising two steps: training some ranking function sorting instances in each class and defining the optimal number of predictions for it. This paper is devoted to the second step of the optimal threshold selection while maximizing the F-macro measure. To do so, we reduce the multi-dimensional problem to the two-dimensional problem of finding a fixed point of a specifically introduced transformation defined on a unit square. We suggest the algorithm of finding the vector of optimal thresholds based on the domain analysis of the introduced transformation. Moreover, we provide the complexity estimations of the proposed algorithm. We evaluate the algorithm on the extreme classification benchmark WikiLSHTC-325K comparing its performance with some baseline results.**

I. Introduction

The growing amount of text and image databases increases the importance of multi-label classification which significantly reduces manual labour and allows the automatic processing of these databases. Classification problems may relate to various areas: biology[1], [2], natural language processing (namely, sentiment classification [3], [4], toxicity classification[5] and text analysis[6]), cybersecurity[7], aerial image processing[8] and even driver drowsiness detection[9].

The problem of the multi-label classification can be divided into two substeps. During the first inference step, some binary classifier for each class (in other words, a ranking function) sorts the instances according to some scores. In practice, at this step one usually employs some probability estimation procedure or even just sorts instances, optimizing some ranking loss, like in extreme classification methods [10]. At the subsequent step, we decide on the number of instances to be assigned to each class so as to improve the score of the chosen evaluation measure. This approach is called plug-in and it has a number of advantages [11], among which there is a possibility to optimize a wide variety of evaluation measures. Another important feature of the plug-in approach is its consistency in the sense of converging to the Bayes optimal prediction [12].

The papers [13], [14], [15], [16] are devoted to the optimal threshold selection for multi-label classification, while a more detailed review can be found in [17]. In general, there are different approaches to the process of threshold selection: rank-based thresholding, proportion-based assignments and score-based local optimization. All of these methods were employed in the works accomplished by [18], [19]. While the first two methods provide only the approximate threshold values, the last one results in an optimal value in a coordinate-wise sense, when the function considered as the function of each threshold separately is optimal [14]. Unfortunately, it is not true for the measure that we put under the analysis in this work.

In the scope of this work, we are concerned with the F-measure optimization. In the literature, there are two approaches to averaging when calculating F-measure for multi-label tasks [20]. According to the first of the definitions, we average the F-measure computed for each class (so we can refer to it as to macro-F measure), while using another one we calculate F-measure as a geometric mean of the average precision and recall (we will call it F-macro vice versa). The former approach is more common, but the latter is also widely employed in many works (for example, in [21], [22], [23]). When precision and recall are balanced in most of the classes, there is no substantial difference between these scores. But given the majority of unbalanced classes, the macro-F becomes significantly smaller than the F-macro score of the average precision and recall. Besides, not only are the optimal threshold values for each class different, but also they lead to dramatically distinct classification results. Precision and recall are included in the macro-F measure in a way that both of them should be high in order to reach the optimum. Meanwhile, computing F-macro allows us to optimize only precision at the expense of low recall in some classes and vice versa in others.

The issues described above make the optimization of the number of recommendations for the F-macro measure a challenging problem as the threshold chosen in one class affects the threshold choice in others. In order to define the vector of thresholds which maximizes the F-macro measure, one can apply the coordinate descent method, though selecting the best threshold in each class separately does not yield an optimal solution. To find the global optimum, we reduce the problem to the problem of finding the fixed point of a two-dimensional transformation $\boldsymbol{V}$ defined on a unit square. After that, we analyse the domain of the resulting transformation and design an algorithm to find out the optimal vector of thresholds. This domain has a complex discrete structure and it requires some heuristics to build a good representation of it. Moreover, we estimate the complexity of the suggested

algorithm.

The developed algorithm of F-macro measure optimization is further applied to the text classification WikiLSHTC-325K dataset [24]. Text classification problems are of great importance nowadays as confirmed by the number of publications concerning text data (e.g. [25], [26], [27]). WikiLSHTC-325K is one of the labelled sets used for benchmarking extreme multi-label classification problems [28] in which the number of classes is immense (325 056 in this particular case). As for the ranking function, we employed the propensity scored, reranked PfastreXML method [29] which can be used on a regular desktop computer. It uses trees to learn the hierarchy of labels and optimizes the nDCG-based ranking loss function which is sensitive to both ranking and relevance.

The remainder of the paper is structured as follows. In Section II we set up the problem and introduce the main definitions. In Section III we reduce the optimization problem to the problem of locating the fixed point of an introduced transformation $\boldsymbol{V}$ and define its domain. In Section IV we demonstrate the suggested algorithm, discuss the motivation behind it and provide some complexity analysis. Section V is devoted to the application of the discussed algorithm to the extreme text classification problem. Section VI discusses the obtained results and elaborates on the possible directions for future work.

II. Problem setting

We consider the problem of a multi-label classification for n intersecting classes c_1, c_2, ..., c_n; $c_k \subset \mathbb{X}$, where $\mathbb{X}$ is the set of objects. Let $\mathbb{Y} = \{0,1\}^n$, and for each $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{X} \times \mathbb{Y}$, the label coordinates $\boldsymbol{y}$ are $y_k = [\![\boldsymbol{x} \in c_k]\!]$ i.e. $y_k = 1$, if $\boldsymbol{x} \in c_k$, and $y_k = 0$ otherwise. A multi-label classifier $\boldsymbol{h} : \mathbb{X} \to \mathbb{Y}$, $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_n(\boldsymbol{x}))$ is essentially a mapping between $\mathbb{X}$ and $\mathbb{Y}$.

The classification quality is evaluated on a test set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$ with the help of F-measure which is the harmonic mean between precision and recall:

$$F(p,r) = \frac{2pr}{p+r}.$$

In each class, precision and recall of a given classifier are computed with the following formulas:

$$p_k(h_k) = \frac{\sum\limits_{i=1}^m y_{ik} h_k(\boldsymbol{x}_i)}{\sum\limits_{i=1}^m h_k(\boldsymbol{x}_i)}, \quad r_k(h_k) = \frac{\sum\limits_{i=1}^m y_{ik} h_k(\boldsymbol{x}_i)}{\sum\limits_{i=1}^m y_{ik}}$$

We employ a specific version of F-measure of macro precision P and recall R calculated as:

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

Throughout the paper, we refer to it as F-macro as macro averaging is performed before the harmonic mean computing.

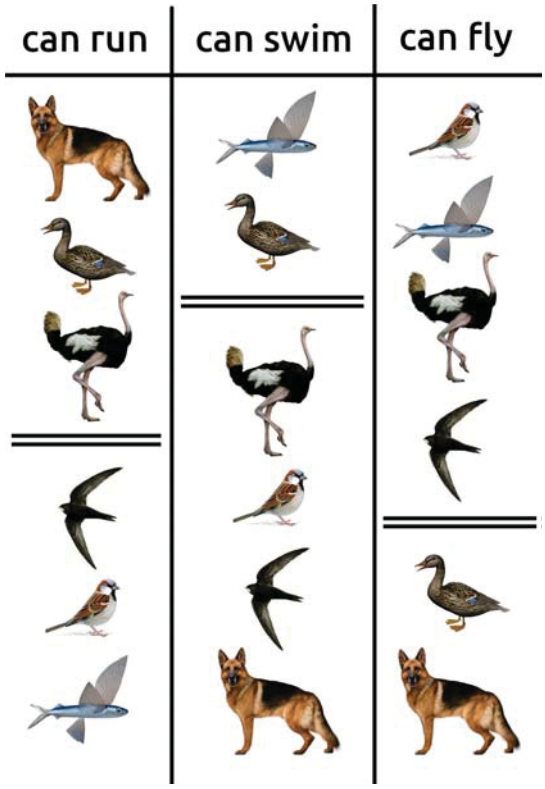

Fig. 1. A simple example of a multi-label classification problem with 3 intersecting classes and 6 objects: a dog, a duck, a flying fish, an ostrich, a sparrow and a swift. The threshold positions are fixed by some given classifier and set to $t_1 = 3, t_2 = 2, t_3 = 4$. The computed F-macro measure for this threshold positions equals to $^{1463}/_{1788}$.

Let us illustrate and clarify the problem setting with a simple example which considers a multi-label classification problem for 3 intersecting classes: „can run“, „can swim“ and „can fly“ (see Fig. 1). There are 6 objects to classify: a dog, a duck, a flying fish, an ostrich, a sparrow and a swift. There also exists a fixed ranking function built with some algorithm which is out of the scope of this article. It sorts the objects in every class according to some probability of their belonging to the class, maybe making some mistakes, and our goal is to select the optimal threshold for each maximizing the F-macro-measure.

Let us compute the F-macro measure for some selected threshold positions: $t_1 = 3, t_2 = 2, t_3 = 3$. For the class „can run“ $p_1 = 1, r_1 = {}^3/_4$ as a sparrow can also run and it was not included into the prediction; for the class „can swim“ $p_2 = 1, r_2 = {}^2/_3$ as a dog can as well swim but it was not included; and for the class „can fly“ $p_2 = {}^3/_4, r_2 = {}^4/_5$ as an ostrich cannot fly while a duck indeed can. Computing the average precision and recall of the given classifier, we obtain $P = {}^{11}/_{12}, R = {}^{133}/_{180}$ which leads to the F-macro measure value of $^{1463}/_{1788}$.

F-macro is monotonic with respect to p_k and r_k, and that is why the optimal classifier in population utility sense $\boldsymbol{h}^*$ indeed has a thresholded form[20]. Therefore, the plug-in approach can be employed for solving this problem.

So, as demonstrated by the example, our goal is to select

the optimal vector of thresholds $\boldsymbol{T} = \{t_k\}_{k=1}^n$ which maximizes the F-macro-measure for the fixed ranking function $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_k) : \mathbb{X} \to [0,1]^n$.

For the sake of brevity, here and after we omit the explicit notation of the threshold classifier $\boldsymbol{h}$ when referring to functions F-macro$(\boldsymbol{h}(\boldsymbol{T}))$, $p_k(h_k(\boldsymbol{T}))$, $r_k(h_k(\boldsymbol{T}))$, $P(\boldsymbol{h}(\boldsymbol{T}))$, $R(\boldsymbol{h}(\boldsymbol{T}))$, which depend on $\boldsymbol{T}$ through $\boldsymbol{h}$, and we will denote them with the same letters as F-macro$(\boldsymbol{T})$, $p_k(t_k)$, $r_k(t_k)$, $P(\boldsymbol{T})$, $R(\boldsymbol{T})$. Using η_k, we can sort the instances in each class and choose the optimal number of observations to be classified as positive. Later on, instead of employing absolute values of η_k, we will use the numbers as thresholds instead. The number $t_k = 1$ corresponds to the threshold position between the first and the second instance in the ranked list, $t_k = 2$ — between the second and the third, etc. The last possible threshold position is $t_k = m$, where m — is the number of instances in the given test set.

III. Reduction to two-dimensional problem

To find the optimum, one can apply the coordinate descent and determine the best threshold in each class one by one. Due to the structure of F-macro, selecting the thresholds in each class independently does not lead to an optimal solution. But in the obtained point for each class c_k the function $F(P(\boldsymbol{T}), R(\boldsymbol{T}))$ (1) as a univariate function of the threshold t_k reaches its maximum. Let $\boldsymbol{T}_0 = (t_1^0, t_2^0, \ldots, t_n^0)$, $\boldsymbol{T}_0^{k,\tau}$ be obtained from $\boldsymbol{T}_0$ by replacing the k-th coordinate by τ:

$$\boldsymbol{T}_0^{k,\tau} = (t_1^0, \ldots, t_{k-1}^0, \tau, t_{k+1}^0, \ldots, t_n^0).$$

Definition 1: We call $\boldsymbol{T}_0$ a coordinate maximum if and only if $\forall k = 1, \ldots, n$ the function $F(P(\boldsymbol{T}_0^{k,\tau}), R(\boldsymbol{T}_0^{k,\tau}))$ of τ reaches its maximum when $\tau = t_k^0$.

All maxima of any function are its coordinate maxima. The opposite is, generally speaking, not true. Taking this into account, one can suggest the following approach to the optimal threshold selection. We need to find all coordinate-wise maxima and choose the one with the biggest F-measure value. Let us find all coordinate maxima of $F(P(\boldsymbol{T}), R(\boldsymbol{T}))$.

Therefore, we decompose our problem and reduce it to a coordinate-wise optima search performed in each class independently. Though this approach does not lead directly to the problem solution, for F-macro given the initial distribution of thresholds $\boldsymbol{T_0}$ we obtain a new one $\boldsymbol{T_1}$ upon which we can construct the following $\boldsymbol{T_2}$, then $\boldsymbol{T_3}$ and so forth.

Definition 2: We define the transformation $\boldsymbol{W}$ of the threshold space as

$$\begin{aligned} &\boldsymbol{W}(P(\boldsymbol{T}), R(\boldsymbol{T})) = \boldsymbol{W}(\boldsymbol{T}) = (w_1(\boldsymbol{T}), w_2(\boldsymbol{T}), ..., w_n(\boldsymbol{T}))\,, \\ &\text{where } w_k(\boldsymbol{T}) = \arg\max_{\tau} F(P(\boldsymbol{T}^{k,\tau}), R(\boldsymbol{T}^{k,\tau})), \\ &\boldsymbol{T}^{k,\tau} = (t_1, \ldots, t_{k-1}, \tau, t_{k+1}, \ldots, t_n). \end{aligned} \tag{2}$$

The coordinate-wise maximum $\boldsymbol{T}$ of $F(P(\boldsymbol{T}), R(\boldsymbol{T}))$ is a fixed point of a threshold space transformation W. Functions $w_k(\boldsymbol{T})$ are, generally speaking, multivalued. In this case, we define $\boldsymbol{T}$ as a fixed point if $\boldsymbol{T}$ belongs to a set of values $\boldsymbol{W}(\boldsymbol{T})$, which we will further denote by the same designation $\boldsymbol{W}(\boldsymbol{T})$.

Locating the fixed point of $\boldsymbol{W}$ is a complicated problem because of the high dimensionality of the threshold space as this dimensionality is equal to the number of classes n. To simplify the process, we examine the countertype of $\boldsymbol{W}$ — a transformation $\boldsymbol{V}$ defined on the square $[0;1]^2$ so as the equality of compositions holds: $\boldsymbol{V} \circ (P, R) = (P, R) \circ \boldsymbol{W}$.

Definition 3: Let P, R be the values of macro precision and recall for some vector of thresholds $\boldsymbol{T}$. Define $\boldsymbol{V}(P, R) = (P(\boldsymbol{W}(\boldsymbol{T})), R(\boldsymbol{W}(\boldsymbol{T})))$.

As well as $\boldsymbol{W}$, $\boldsymbol{V}$ is also multivalued. In this case, the precision P and recall R of a set of threshold vectors $\boldsymbol{W}(\boldsymbol{T})$ are calculated as images under the functions P and R correspondingly. Let us find the fixed points of $\boldsymbol{V}$.

The domain of the transformation $\boldsymbol{V}$ has a complicated discrete structure. It can be defined as:

$$\begin{aligned} \mathcal{D}(\boldsymbol{V}) = \Big\{ &(P, R) \mid \exists \boldsymbol{T}\ P = P(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{n} p_k(t_k), \\ &R = R(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{n} r_k(t_k) \Big\} \end{aligned} \tag{3}$$

Each point $(P, R) \in \mathcal{D}(\boldsymbol{V})$ is an average value of the precision p_k and recall r_k for some vector of thresholds $\boldsymbol{T}$.

IV. Algorithm

If there is a good approximation to the domain $\mathcal{D}(\boldsymbol{V})$, then maximizing $F(P, R)$ on it allows us to find the optimal values of precision P and recall R, by which one can define the optimal vector of thresholds.

Even if the problem is not high-dimensional, for example, the number of instances is $m = 1000$ and the number of classes is $n = 10$, the number of points comprised in $\mathcal{D}(\boldsymbol{V})$ is about 1000^{10} and, therefore, infeasible to process. But we can avoid overflowing by tackling this issue from another side.

We consider the convex hull of $\mathcal{D}(V)$ $H = \text{hull}\,\mathcal{D}(V)$. As $F(P, R)$ defined on $[0;1]^2$ does not have internal maximums in $[0;1]^2$, then the maximum in H is attained at the boundary H. This way we can approximately obtain the maximum of $F(P, R)$ on $\mathcal{D}(V)$. Theorem 1 allows us to reduce the number of points under consideration and advocates for a new approach to find the optimal solution to the problem.

Theorem 1: Let $\Gamma \subset \mathcal{D}(V)$ be the set of H vertices. Then

$$\max_{\Gamma} F \le \max_{\mathcal{D}(V)} F \le \max_{H} F.$$

The proof results directly from the embedding

$$\Gamma \subset \mathcal{D}(V) \subset H.$$

A straightforward maximization of a two-variable function $F(P, R)$ on the convex polygon H and a search for the vertex from Γ with a minimal function F value provide us with the upper and lower estimations for the unknown maximum of F on $\mathcal{D}(V)$. The extremum vertex from Γ represents a vector of thresholds close to the optimal one.

The described procedure allows us to suggest Algorithm 1. It subsequently computes the approximation H_i to H

$$H_i = \text{hull}\Big\{(P,R) \mid \exists \boldsymbol{T}\ P = P(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{i} p_k(t_k), \\ R = R(\boldsymbol{T}) = \frac{1}{n}\sum_{k=1}^{i} r_k(t_k)\Big\}, \quad i = 1,2,\ldots,n. \tag{4}$$

H_i are easily calculated with the following recursive formula

$$H_{i+1} = \text{hull}\left(H_i + \text{hull}(\{p_i(\tau), r_i(\tau)\}_{\tau=1}^{m})\right). \tag{5}$$

According to the central limit theorem, the average values of precision and recall in the classes are distributed according to the Gaussian law. The number of points in $\mathcal{D}(V)$ grows not faster than $N = m^n$. As mentioned in [30], the number of vertices in a convex hull of normally distributed samples grows not faster than $2\sqrt{2\pi \ln N} = O(\sqrt{n \ln m})$. Using the described Algorithm 1, at each step we compute the convex hull, add new points to consideration and then, compute another convex hull. Given that linear time is required for the convex hull computation, overall time computational complexity is $O(nm\sqrt{n \ln m})$. So, with Algorithm 1 we manage to improve complexity comparing to direct computation of convex hull of $\mathcal{D}(V)$, but we can do better.

Algorithm 1 Construction of $\text{hull}\mathcal{D}(V)$

1: **Input**: lists of instances $\boldsymbol{x}$, sorted in each class c_i, $i = 1..n$ according to some fixed ranking function $\boldsymbol{\eta(x)}$ and true class membership $\boldsymbol{y}$.
2: $H_1 = \text{hull}(\{p_1(\tau), r_1(\tau)\}_{\tau=1}^{m})$
3: **for** $i = 2,\ldots,n$ **do**
4: $$H_i = \text{hull}\left(H_{i-1} + \text{hull}(\{p_i(\tau), r_i(\tau)\}_{\tau=1}^{m})\right)$$
5: **end for**
6: Find the point $(P^*_\Gamma, R^*_\Gamma) \in \Gamma$ in which F-macro reaches its maximum — the lower bound for maximum of F-macro on $\mathcal{D}(V)$.
7: Find the point $(P^*_H, R^*_H) \in H$ in which F-macro reaches its maximum — the upper bound for maximum of F-macro on $\mathcal{D}(V)$.
8: Compute the optimal vector of thresholds $\boldsymbol{T}^* = \boldsymbol{W}(P^*_\Gamma, R^*_\Gamma)$ corresponding to (P^*_Γ, R^*_Γ).

Given the polar angle ϕ, we denote by $e(\phi)$ the most distant point of $\mathcal{D}(V)$ in this direction

$$e(\phi) = \underset{(p,r)\in\mathcal{D}(V)}{\arg\max}\ (p\cos\phi + r\sin\phi).$$

Consider the grid of n_α different directions (polar angles). Let us limit the computations of H vertices to n_α edge points the most distant in these directions. Again we examine each category and compute the convex hull $\text{hull}(\{p_i(\tau), r_i(\tau)\}_{\tau=1}^{m})$ for each category. But in this modification of the algorithm represented as Algorithm 2, we recount only some of n_α points on the convex hull. The greater n_α, the more precise the approximation to the strict convex hull is. The time complexity for Algorithm 2 is much improved and now makes $O(n_\alpha nm)$.

Algorithm 2 Fast construction of $\text{hull}\mathcal{D}(V)$

1: **Input**: lists of instances $\boldsymbol{x}$, sorted in each class c_i, $i = 1..n$, according to some fixed ranking function $\boldsymbol{\eta(x)}$ and true class membership $\boldsymbol{y}$, number of polar angles n_α.
2: Initialize the approximation to convex hull randomly $\Gamma_\alpha = \{(0,0)\}_{j=1}^{n_\alpha}$
3: **for** $i = 1,\ldots,n$ **do**
4: $M_i = \text{hull}(\{p_i(\tau), r_i(\tau)\}_{\tau=1}^{m})$
5: Iterate over points in M_i in counter-clockwise order:
6: **for** (p_m, r_m) in M_i **do**
7: Compute polar angles ϕ_{prev} and ϕ_{next} of normals to previous segment between (p_{m-1}, r_{m-1}) and (p_m, r_m) and next segment between (p_m, r_m) and (p_{m+1}, r_{m+1}).
8: Update vertices $e(\phi) = (p_\phi, r_\phi) \in \Gamma_\alpha$ for $\phi \in [\phi_{prev}, \phi_{next}]$:
$$p_\phi = \frac{p_\phi \cdot (i-1) + p_m}{i}, \quad r_\phi = \frac{r_\phi \cdot (i-1) + r_m}{i}.$$
9: **end for**
10: **end for**
11: Find the point $(P^*, R^*) \in \Gamma_\alpha$ in which F-macro reaches its maximum — the lower bound for maximum of F-macro on $\mathcal{D}(V)$.
12: Compute the optimal vector of thresholds $\boldsymbol{T}^* = \boldsymbol{W}(P^*, R^*)$ corresponding to (P^*, R^*).

V. Experiments

In our experiments we consider a dataset used as a benchmark in extreme multi-label learning where the goal is to learn features and classifiers that can automatically tag the instance with the most relevant subset of labels from an extremely large labelled set. The WikiLSHTC-325K dataset contains 2 365 435 documents and is annotated with $n = 325\,056$ classes representing one of extreme classification problems (see more in the repository [28]). The training set employed during the experiments comprises 1 778 351 documents, the test set consists of $m = 587\,084$ documents. Fig. 2 exhibits the distribution of classes in the test set, demonstrating that the majority of classes are very small and contain less than 20 documents which makes the problem more perplex.

To obtain the scores for the test instances in a reasonable time, we apply the propensity scored reranked PfastreXML method [29]. By default, it predicts 5 labels per instance and we report the results for these scores in the last column of Table I. Still, we consider these results insufficient and, therefore, increase the number of predicted labels per instance to achieve better results reported in the second column of Table I.

To investigate the classifier performance in different classes, we build the precision-recall curves for some of them (shown in Fig. 3) which demonstrate the trade off in performance between precision and recall for different values of threshold.

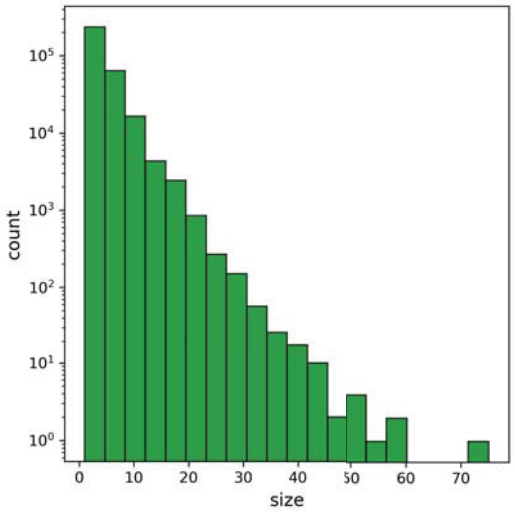

Fig. 2. Class sizes distribution for the WikiLSHTC-325K dataset, x-axis represents size of the classes, y-axis (given in log scale) stands for the number of classes of this size.

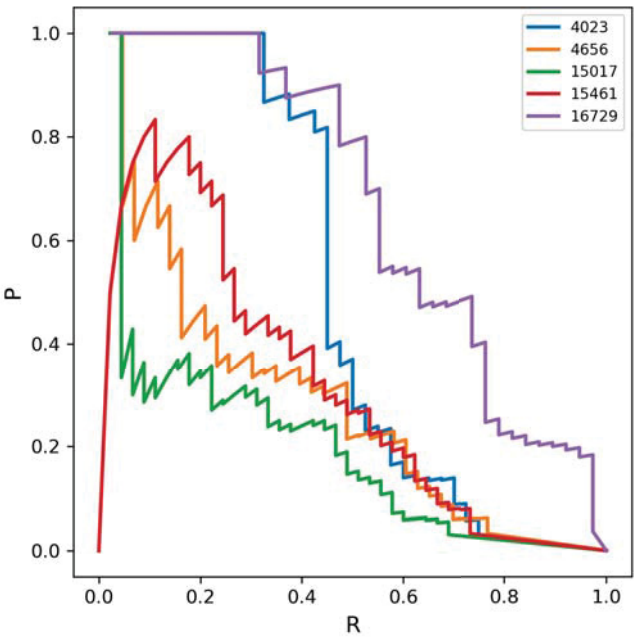

Fig. 3. Precision-recall curves for the categories 4023, 4656, 15017, 15461 and 16729 from the WikiLSHTC-325K dataset and the classifier built with PfastreXML method. The choice of the categories is random among the categories containing at least 30 documents for precision-recall curves to look smoother.

The categories are chosen randomly but so as their size extends 30 documents for precision-recall curves to look smoother. The better the classifier performs in the class, the closer the curve bows to the point with coordinates (1, 1). The best threshold for each class should be selected close to the threshold corresponding to the point closest to the upper right corner of the unit square.

To compare the optimal F-macro values, we use the following baseline. First, we optimize the F measure of each class independently and calculate corresponding threshold values $T^b = (t_1^b, \ldots, t_n^b)$:

$$t_k^b = \arg\max_{\tau} F(p_k(\tau), r_k(\tau)), \quad k = 1, \ldots, n. \qquad (6)$$

Thus, we maximize macro-F measure: $\max_T \text{macro-F}(T) = \text{macro-F}(T^b)$. Then we evaluate F-macro and macro-F for same threshold vector T^b. The resulting values are presented in Table I.

TABLE I. F-MEASURE VALUES FOR THE TEST PART OF THE WIKILSHTC-325K DATASET. THE $F(P, R)$ VALUE STANDS FOR THE RESULT OF THE PROPOSED ALGORITHM 2. THE BASELINE VALUES F-macro(T^b) AND macro-F(T^b) ARE OBTAINED FOR THE THRESHOLDS T^b WHICH MAXIMIZE F MEASURE IN EACH CLASS INDEPENDENTLY.

	WikiLSHTC 1000 lpi	WikiLSHTC 5 lpi
$F(P, R)$	**0.693**	**0.439**
F-macro(T^b)	0.654	0.422
macro-F(T^b)	0.533	0.206

We also demonstrate the appearance of $\mathcal{D}(V)$ built for the WikiLSHTC-325K dataset in Fig. 4. One can observe the discrete structure of the domain which sophisticates the process of its building. The optimal point corresponding for the highest F-macro measure is equal to $(0.631, 0.768)$.

VI. CONCLUSION

When solving multi-label classification problems, F-measure of macro precision and recall is especially difficult for optimization as it does not allow separate parallel optimization in each class. Cyclic optimization procedure provides us with some local coordinate-wise maximum which is not guaranteed to be a global optimum. Therefore, the need in simple and efficient solution is indeed strong and vast.

The proposed method is based on introducing a transformation and the further analysis of its domain. As the resulting transformation is two-dimensional, it is already much more pleasant to deal with. Moreover, we suggest the approach allowing us to obtain the point in which the F-measure reaches its optimum. The complexity of the suggested algorithm depends on the chosen algorithm of the convex hull building which can be done in linear time with respect to the number of vertices. It also exploits the idea of building a fast approximation to the convex hull which allows us to evaluate it in a reasonable time. The overall complexity of the algorithm is estimated as $O(n_\alpha nm)$ as we need to look through all elements in every category to obtain all the possible domain points.

The suggested approach is applied to one of the benchmark extreme multi-label text classification datasets — WikiLSHTC-325K. The work of the suggested algorithm is demonstrated and the results obtained with the means of this algorithm are compared with some baseline results, clearly demonstrating the advantage of the suggested approach.

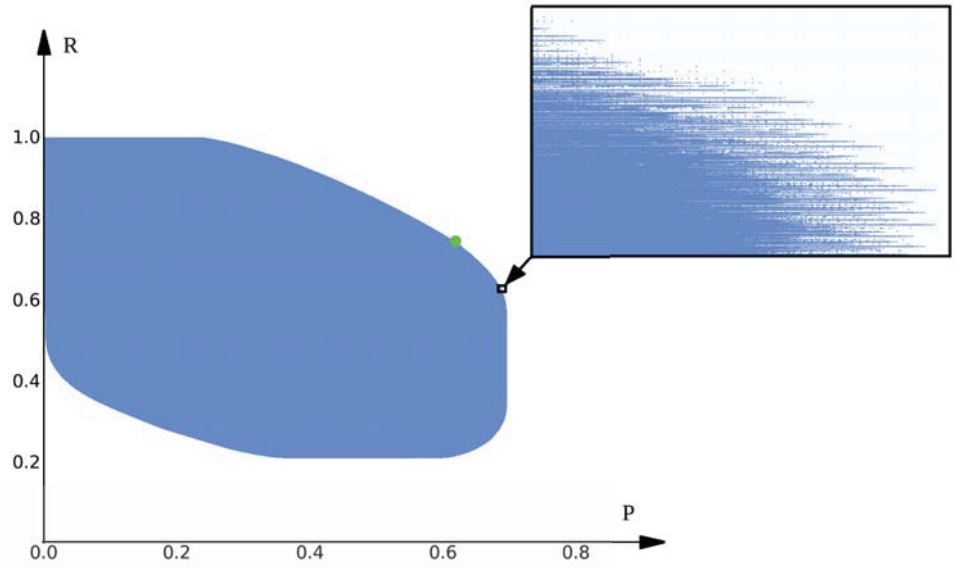

Fig. 4. The domain of V for the test set of the WikiLSHTC-325K dataset. The green point corresponds to the optimal values of P and R.

There are some possible directions in which this work may progress. The process of building the convex hull can be improved if the convex hull is built on-the-fly upon the considering each class. Another conceivable problem to tackle is to investigate the applicability of this method to optimization of different widely used quality metrics.

ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research project № 20-37-90038.

REFERENCES

[1] G. Kolokolnikov and A. Samorodov, "Comparative study of data augmentation strategies for white blood cells classification," in *2019 25th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019, pp. 168–175.

[2] D. Korotaeva, M. Khlopotov, A. Makarenko, E. Chikshova, N. Startseva, and A. Chemysheva, "Botanicum: a telegram bot for tree classification," in *2018 22nd Conference of Open Innovations Association (FRUCT)*. IEEE, 2018, pp. 88–93.

[3] K. Lagutina, V. Larionov, V. Petryakov, N. Lagutina, and I. Paramonov, "Sentiment classification of russian texts using automatically generated thesaurus," in *2018 23rd Conference of Open Innovations Association (FRUCT)*. IEEE, 2018, pp. 217–222.

[4] K. Lagutina, V. Larionov, V. Petryakov, N. Lagutina, I. Paramonov, and I. Shchitov, "Sentiment classification into three classes applying multinomial bayes algorithm, n-grams, and thesaurus," in *2019 24th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019, pp. 214–219.

[5] S. Morzhov, "Avoiding unintended bias in toxicity classification with neural networks," in *2020 26th Conference of Open Innovations Association (FRUCT)*. IEEE, 2020, pp. 314–320.

[6] K. Lagutina, N. Lagutina, E. Boychuk, and I. Paramonov, "The influence of different stylometric features on the classification of prose by centuries," in *2020 27th Conference of Open Innovations Association (FRUCT)*. IEEE, 2020, pp. 108–115.

[7] A. A. Vorobeva, "Influence of features discretization on accuracy of random forest classifier for web user identification," in *2017 20th Conference of Open Innovations Association (FRUCT)*. IEEE, 2017, pp. 498–504.

[8] D. Kasimov, A. Kuchuganov, and V. Kuchuganov, "Methods and tools for developing decision rules for classifying objects in aerial images," in *2020 26th Conference of Open Innovations Association (FRUCT)*. IEEE, 2020, pp. 158–165.

[9] A. Kashevnik, K. Karelskaya, and M. Repp, "Dangerous situations determination by smartphone in vehicle cabin: Classification and algorithms," in *2019 24th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019, pp. 130–139.

[10] Y. Prabhu and M. Varma, "Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2014.

[11] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent multilabel classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 3321–3329.

[12] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier, "Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization," in *International Conference on Machine Learning*, 2013, pp. 1130–1138.

[13] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier, "Extreme f-measure maximization using sparse probability estimates," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, pp. 1435–1444. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045390.3045542

[14] I. Pillai, G. Fumera, and F. Roli, "Threshold optimisation for multi-label classifiers," *Pattern Recogn.*, vol. 46, no. 7, pp. 2055–2065, Jul. 2013.

[15] Y. Yang, "A study of thresholding strategies for text categorization," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '01. New York, NY, USA: ACM, 2001, pp. 137–145. [Online]. Available: http://doi.acm.org/10.1145/383952.383975

[16] R.-E. Fan and C.-J. Lin, "A study on threshold selection for multi-label classification," *Department of Computer Science, National Taiwan University*, pp. 1–23, 2007.

[17] I. Pillai, G. Fumera, and F. Roli, "Designing multi-label classifiers that maximize f measures," *Pattern Recogn.*, vol. 61, no. C, pp. 394–404, Jan. 2017.

[18] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005332.1005345

[19] I. Triguero and C. Vens, "Labelling strategies for hierarchical multi-label classification techniques," *Pattern Recogn.*, vol. 56, no. C, pp. 170–183, Aug. 2016.

[20] A. Berger and S. Guda, "Threshold optimization for f measure of macro-averaged precision and recall," *Pattern Recognition*, p. 107250, 2020.
[21] M. Cornolti, P. Ferragina, and M. Ciaramita, "A framework for benchmarking entity-annotation systems," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: ACM, 2013, pp. 249–260. [Online]. Available: http://doi.acm.org/10.1145/2488388.2488411
[22] L. Luo and L. Li, "Defining and evaluating classification algorithm for high-dimensional data based on latent topics," in *PloS one*, 2014.
[23] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A LSTM based framework for handling multiclass imbalance in DGA botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, 2018.
[24] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artières, G. Paliouras, É. Gaussier, I. Androutsopoulos, M. Amini, and P. Gallinari, "LSHTC: A benchmark for large-scale text classification," *CoRR*, vol. abs/1503.08581, 2015.
[25] N. S. Lagutina, K. V. Lagutina, I. A. Shchitov, and I. V. Paramonov, "Analysis of influence of different relations types on the quality of thesaurus application to text classification problems," *Modeling and Analysis of Information System*, vol. 24, no. 6, pp. 772–787, 2017.
[26] F. Gargiulo, S. Silvestri, M. Ciampi, and G. De Pietro, "Deep neural network for hierarchical extreme multi-label text classification," *Applied Soft Computing*, vol. 79, pp. 125–138, 2019.
[27] S. V. Morzhov, "Modern approaches to detect and classify comment toxicity using neural networks," *Modeling and Analysis of Information Systems*, vol. 27, no. 1, pp. 48–61, 2020.
[28] M. Varma. (2018) The extreme classification repository: Multi-label datasets & code. [Online]. Available: http://manikvarma.org/downloads/XC/XMLRepository.html
[29] H. Jain, Y. Prabhu, and M. Varma, "Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 935–944.
[30] I. Hueter, "The convex hull of a normal sample," *Advances in applied probability*, vol. 26, no. 4, pp. 855–875, 1994.