Aggressive Sampling for Multi-class to Binary Reduction with Applications to Text Classification

Bikash Joshi
Univ. Grenoble Alps, LIG
Grenoble, France
bikash.joshi@imag.fr

Massih-Reza Amini
Univ. Grenoble Alps, LIG
Grenoble, France
massih-reza.amini@imag.fr

Ioannis Partalas
Expedia EWE
Geneva, Switzerland
ipartalas@expedia.com

Franck Iutzeler
Univ. Grenoble Alps, LJK
Grenoble, France
franck.iutzeler@imag.fr

Yury Maximov
Los Alamos National Laboratory and Skolkovo IST,
USA and Moscow, Russia
yury@lanl.gov

Abstract

We address the problem of multi-class classification in the case where the number of classes is very large. We propose a double sampling strategy on top of a multi-class to binary reduction strategy, which transforms the original multi-class problem into a binary classification problem over pairs of examples. The aim of the sampling strategy is to overcome the curse of long-tailed class distributions exhibited in majority of large-scale multi-class classification problems and to reduce the number of pairs of examples in the expanded data. We show that this strategy does not alter the consistency of the empirical risk minimization principle defined over the double sample reduction. Experiments are carried out on DMOZ and Wikipedia collections with 10,000 to 100,000 classes where we show the efficiency of the proposed approach in terms of training and prediction time, memory consumption, and predictive performance with respect to state-of-the-art approaches.

1 Introduction

Large-scale multi-class or extreme classification involves problems with extremely large number of classes as it appears in text repositories such as Wikipedia, Yahoo! Directory (www.dir.yahoo.com), or Directory Mozilla DMOZ (www.dmoz.org); and it has recently evolved as a popular branch of machine learning with many applications in tagging, recommendation and ranking. The most common and popular baseline in this case is the one-versus-all approach (OVA) [18] where one independent binary classifier is learned per class. Despite its simplicity, this approach suffers from two main limitations; first, it becomes computationally intractable when the number of classes grow large, affecting at the same time the prediction. Second, it suffers from the class imbalance problem by construction. Recently, two main approaches have been studied to cope with these limitations. The first one, broadly divided in tree-based and embedding-based methods, have been proposed with the aim of reducing the effective space of labels in order to control the complexity of the learning problem. Tree-based methods [4, 6, 7, 9, 21, 5, 15] rely on binary tree structures where each leaf corresponds to a class and inference is performed by traversing the tree from top to bottom; a binary classifier being used at each node to determine the child node to develop. These methods have logarithmic time complexity with the drawback that it is a challenging task to find a balanced tree structure which can partition the class labels. Further, even though different heuristics have been developed to address the unbalanced problem, these methods suffer from the drawback that they have to make several decisions prior to reaching a final category, which leads to error propagation and
2 A doubly-sampled multi-class to binary reduction strategy

We address the problem of monolabel multi-class classification defined on joint space \( X \times Y \) where \( X \subseteq \mathbb{R}^d \) is the input space and \( Y = \{1, \ldots, K\} \) the output space, made of \( K \) classes. Elements of \( X \times Y \) are denoted as \( x^y = (x, y) \). Furthermore, we assume the training set \( S = (x^y)^m \) is made of \( m \) i.i.d examples/class pairs distributed according to a fixed but unknown probability distribution \( D \), and we consider a class of predictor functions \( G = \{ g : X \times Y \to \mathbb{R} \} \).

We define the instantaneous loss for predictor \( g \in G \) on example \( x^y \) as:

\[
e(g, x^y) = \frac{1}{K - 1} \sum_{y' \in Y \setminus \{y\}} \mathbb{1}_{g(x^y) \leq g(x^{y'})},
\]

where \( \mathbb{1} \) is the indicator function equal to 1 if the predicate \( \pi \) is true and 0 otherwise. Compared to the classical multi-class error, \( e'(g, x^y) = \mathbb{1}_{y \neq \arg \max_{y' \in Y} g(x^{y'})} \), the loss of (1) estimates the average number of classes, given any input data, that get a greater scoring by \( g \) than the correct class. The loss (1) is hence a ranking criterion, and the multi-class SVM of [29] and AdaBoost.MR [24] optimize convex surrogate functions of this loss. It is also used in label ranking [12]. Our objective is to find a function \( g \in G \) with a small expected risk \( R(g) = \mathbb{E}_{x^y \sim D} [e(g, x^y)] \), by minimizing the empirical error defined as the average number of training examples \( x^{y_i} \in S \) which, in mean, are scored lower than \( x^y \), for \( y' \in Y \setminus \{y_i\} \):

\[
\hat{R}_m(g, S) = \frac{1}{m} \sum_{i=1}^{m} e(g, x^{y_i}) = \frac{1}{m(K - 1)} \sum_{i=1}^{m} \sum_{y' \in Y \setminus \{y_i\}} \mathbb{1}_{g(x^{y_i}) - g(x^{y'}) \leq 0}.
\]

2.1 Binary reduction based on dyadic representations of examples and classes

In this work, we consider prediction functions of the form \( g = f \circ \phi \), where \( \phi : X \times Y \to \mathbb{R}^p \) is a projection of the input and the output space into a joint feature space of dimension \( p \); and \( f \in F = \{ f : \mathbb{R}^p \to \mathbb{R} \} \) is a function that measures the adequacy between an observation \( x \) and a class \( y \) using their corresponding representation \( \phi(x^y) \). The projection function \( \phi \) is application-dependent and it can either be learned [29], or defined using some heuristics [27, 16].
Further, consider the following dyadic transformation

\[ T(S) = \left\{ \begin{array}{ll} (z_j = (\phi(x^j_k), \phi(x^j_0)), \tilde{y}_j = -1) & \text{if } k < y_j \\ (z_j = (\phi(x^j_k), \phi(x^j_0)), \tilde{y}_j = +1) & \text{elsewhere} \end{array} \right. \]

(3)

where \( j = (i - 1)(K - 1) + k \) with \( i \in [m], k \in [K - 1] \); that expands a \( K \)-class labeled set \( S \) of size \( m \) into a binary labeled set \( T(S) \) of size \( N = m(K - 1) \) (e.g. Figure 1 over a toy problem).

With the class of functions

\[ \mathcal{H} = \{ h : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}; (\phi(x^p), \phi(x^y)) \mapsto f(\phi(x^y)) - f(\phi(x^p)), f \in F \}, \]

(4)

the empirical loss (Eq. (2)) can be rewritten as:

\[ \tilde{R}_{T(S)}(h) = \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{\tilde{y}_j h(z_j) \leq 0}. \]

(5)

Hence, the minimization of Eq. (5) over the transformation \( T(S) \) of a training set \( S \) defines a binary classification over the pairs of dyadic examples. However, this binary problem takes as examples dependent random variables, as for each original example \( x^p \in S \), the \( K - 1 \) pairs in \( \{ (\phi(x^p), \phi(x^y)): \tilde{y} \in T(S) \} \) all depend on \( x^p \). In [16] this problem is studied by bounding the generalization error associated to \( S \) using the fractional Rademacher complexity introduced in [25]. In this work, we derive a new generalization bounds based on Local Rademacher Complexities introduced in [22] that implies second-order (i.e. variance) information inducing faster convergence rates (Theorem 1).

Our analysis relies on the notion of graph covering introduced in [14] and defined as:

**Definition 1** (Exact proper fractional cover of \( G \), [14]). Let \( G = (V, E) \) be a graph. \( C = \{ (C_k, \omega_k) \}_{k \in [J]} \) for some positive integer \( J \), with \( C_k \subseteq V \) and \( \omega_k \in [0, 1] \) is an exact proper fractional cover of \( G \), if: i) it is proper: \( \forall k, C_k \) is an independent set, i.e., there is no connections between vertices in \( C_k \); ii) \( G \) is an exact fractional cover of \( G \): \( \forall v \in V, \sum_{k : v \in C_k} \omega_k = 1 \).

The weight \( W(C) \) of \( C \) is given by: \( W(C) = \sum_{k \in [J]} \omega_k \) and the minimum weight \( \chi^*(G) = \min_{C_k \in K(G)} W(C) \) over the set \( K(G) \) of all exact proper fractional covers of \( G \) is the fractional chromatic number of \( G \). From this statement, [14] extended Hoeffding’s inequality and proposed large deviation bounds for sums of dependent random variables which was the precursor of new generalisation bounds, including a Talagrand’s type inequality for empirical processes in the dependent case presented in [22].

With the classes of functions \( G \) and \( \mathcal{H} \) introduced previously, consider the parameterized family \( \mathcal{H}_r \) which, for \( r > 0 \), is defined as:

\[ \mathcal{H}_r = \{ h : h \in \mathcal{H}, \forall h \Rightarrow \forall z, \mathbb{1}_{\tilde{y} h(z)} \leq r \}, \]

where \( \mathbb{1} \) denotes the variance.

The fractional Rademacher complexity introduced in [25] entails our analysis:

\[ \mathcal{R}_{T(S)}(\mathcal{H}) = \frac{2}{N} \varepsilon \sum_{k \in [K - 1]} \omega_k \mathbb{E}_{C_k} \sup_{h \in \mathcal{H}} \sum_{x \in C_k} \xi_n h(x_n), \]

with \( (\xi_n)_{n=1}^{N} \) a sequence of independent Rademacher variables verifying \( P(\xi_n = 1) = P(\xi_n = -1) = \frac{1}{2} \). If other is not specified explicitly we assume below all \( \omega_k = 1 \). Our first result that bounds the generalization error of a function \( h \in \mathcal{H} \): \( R(h) = \mathbb{E}_{T(S)}[\tilde{R}_{T(S)}(h)] \), with respect to its empirical error \( \tilde{R}_{T(S)}(h) \) over a transformed training set, \( T(S) \), and the fractional Rademacher complexity, \( \mathcal{R}_{T(S)}(\mathcal{H}) \), is stated below.

Figure 1: A toy example depicting the transformation \( T(E) \) applied to a training set \( S \) of size \( m = 4 \) and \( K = 4 \).

Figure 2: The dependency graph \( G = \{ 1, \ldots, 12 \} \) corresponding to the toy problem of Figure 1 where dependent nodes are connected with vertices in blue double-line. The exact proper fractional cover \( C_1, C_2 \) and \( C_3 \) is shown in dashed. The fractional chromatic number is in this case \( \chi^*(G) = K - 1 = 3 \).
After this double sampling, we apply the transformation
we set them inversely proportional to the size of each
class in the original training set; \( \forall k, \pi_k = \frac{1}{\mu_k} \) where \( \mu_k \) is the proportion of class \( k \) in \( S \). The second aim is to reduce the number of dyadic examples controlled by the hyperparameter \( \kappa \). The pseudo-code of this aggressive double sampling procedure, referred to as \((\pi, \kappa)\)-DS, is depicted above and it is composed of two main steps.

1. For each class \( k \in \{1, \ldots, K\} \), draw randomly a set \( S_{\pi_k} \) of examples from \( S \) of that class with probability \( \pi_k \), and let \( S_\pi = \bigcup_{k=1}^{K} S_{\pi_k} \).

2. For each example \( x^y \) in \( S_\pi \), draw uniformly \( \kappa \) adversarial classes in \( \mathcal{Y} \setminus \{y\} \).

After this double sampling, we apply the transformation \( T \) defined as in Eq. (3), leading to a set \( T_\kappa(S_\pi) \) of size \( \kappa|S_\pi| \ll N \).

In Section 2.3 we will show that this procedure practically leads to dramatic improvements in terms of memory consumption, computational complexity, and a higher multi-class prediction accuracy when the number of classes is very large. The empirical loss over the transformation of the new subsampled training set \( S_\pi \) of size \( M \), outputted by the \((\pi, \kappa)\)-DS algorithm is:

\[
\tilde{R}_{T_\kappa(S_\pi)}(h) = \frac{1}{\kappa M} \sum_{(x^y, z^y) \in T_\kappa(S_\pi)} 1_{g(x^y, h(z^y)) \leq 0} = \frac{1}{\kappa M} \sum_{x^y \in S_\pi} \sum_{y' \in \mathcal{Y} \setminus \{y\}} 1_{g(x^y) - g(x^{y'}) \leq 0},
\] (6)

which is essentially the same empirical risk as the one defined in Eq. (2) but taken with respect to the training set outputted by the \((\pi, \kappa)\)-DS algorithm. Our main result is the following theorem which bounds the generalization error of a function \( h \in \mathcal{H} \) learned by minimizing \( \tilde{R}_{T_\kappa(S_\pi)} \).
Furthermore, for all functions in the class $H$ within that class. Note that class centroids are computed once in the preliminary projection of training

Theorem 2. Let $\mathcal{S} = (x_i^m, y_i^m)_{i=1}^m \in (X \times Y)^m$ be a training set of size $m$ i.i.d. according to a probability distribution $\mathcal{D}$ over $X \times Y$, and $T(\mathcal{S}) = ((z_i, \tilde{y}_i))_{i=1}^n$ the transformed set obtained with the transformation function $T$ defined as in Eq. (3). Let $\mathcal{S}_\pi \subseteq \mathcal{S}$, $|\mathcal{S}_\pi| = M$, be a training set outputted by the algorithm $(\pi, \kappa)$-DS and $T(\mathcal{S}_\pi) \subseteq T(\mathcal{S})$ its corresponding transformation. Then for any $1 > \delta > 0$ with probability at least $(1 - \delta)$ the following risk bound holds for all $h \in H$:

$$R(h) \leq \tilde{\alpha}R_{\text{DS}}(\mathcal{S}_\pi)(h) + \alpha \mathcal{R}_{\mathcal{S}_\pi}(\ell \circ H) + \alpha \sqrt{\frac{(K - 1) \log \frac{2}{\delta}}{2M\kappa}} + \sqrt{\frac{2\alpha \log 4K \delta}{\beta(m - 1)}} + \frac{7\beta \log 4K \delta}{3(m - 1)}.$$  

Furthermore, for all functions in the class $\mathcal{H}_\ell$, we have the following generalization bound that holds with probability at least $(1 - \delta)$:

$$R(h) \leq \alpha R_{\text{DS}}(\mathcal{S}_\pi)(h) + \alpha \mathcal{R}_{\mathcal{S}_\pi}(\ell \circ H_\ell) + \sqrt{\frac{2\alpha \log 4K \delta}{\beta(m - 1)}} + \frac{7\beta \log 4K \delta}{3(m - 1)}$$

$$+ 5\alpha \left( \sqrt{\mathcal{R}_{\mathcal{S}_\pi}(\ell \circ H_\ell)} + \frac{\sqrt{r}}{2} \right) \sqrt{\frac{(K - 1) \log \frac{2}{\delta}}{M\kappa}} + \frac{25\alpha \log \frac{2}{\delta}}{48M},$$

where $\ell : \{-1, +1\} \times \mathbb{R} \rightarrow [0, 1] 0/1$ is an instantaneous loss, and $\alpha = \max_{y: 1 \leq y \leq K} \eta_y / \pi_y$, $\beta = \max_{y: 1 \leq y \leq K} 1 / \pi_y$ and $\eta_y > 0$ is the proportion of class $y$ in $\mathcal{S}$.

The proof is provided in the supplementary material. This theorem hence paves the way for the consistency of the empirical risk minimization principle [26, Th. 2.1, p. 38] defined over the double sample reduction strategy we propose.

### 2.3 Prediction with Candidate Selection

The prediction is carried out in the dyadic feature space, by first considering the pairs constituted by a test observation and all the classes, and then choosing the class that leads to the highest score by the learned classifier.

**Algorithm:** Prediction with Candidate Selection Algorithm

**Input:** Unlabeled test set $T$;

**Learned function** $f^* : \mathbb{R}^p \rightarrow \mathbb{R}$;

**Initialization:** $\Omega \leftarrow \emptyset$;

**forall** $x \in T$ **do**

- Select $Y_x \subseteq Y$ candidate set of $q$ nearest-centroid classes;
- $\Omega \leftarrow \Omega \cup \text{argmax}_{y \in Y_x} f^*(\phi(x^y))$;

**return** predicted classes $\Omega$

### 3 Experiments

In this section, we provide an empirical evaluation of the proposed reduction approach with the $(\pi, \kappa)$-DS sampling strategy for large-scale multi-class classification of document collections. First, we present the mapping $\phi : X \times Y \rightarrow \mathbb{R}^p$. Then, we provide a statistical and computational comparison of our method with state-of-the-art large-scale approaches on popular datasets.

**3.1 a Joint example/class representation for text classification**

The particularity of text classification is that documents are represented in a vector space induced by the vocabulary of the corresponding collection [23]. Hence each class can be considered as a mega-document, constituted by the concatenation of all of the documents in the training set belonging to it,

$^1$The number of classes pre-selected can be tuned to offer a prediction time/accuracy tradeoff if the prediction time is more critical.
and simple feature mapping of examples and classes can be defined over their common words. Here we used $p=10$ features inspired from learning to rank \[17\] by resembling a class and a document to respectively a document and a query (Table \[1\]). All features except feature 9, that is the distance of an example $x$ to the centroid of all examples of a particular class $y$, are classical. In addition to its predictive interest, the latter is also used in prediction for performing candidate preselection. Note that for other large-scale multi-class classification applications like recommendation with extremely large number of offer categories or image classification, a same kind of mapping can either be learned or defined using their characteristics \[22\]-\[28\].

### 3.2 Experimental Setup

**Datasets.** We evaluate the proposed method using popular datasets from the Large Scale Hierarchical Text Classification challenge (LSHTC) 1 and 2 \[20\]. These datasets are provided in a pre-processed format using stop-word removal and stemming. Various characteristics of these datasets including the statistics of train, test and heldout are listed in Table \[2\]. Since, the datasets used in LSHTC2 challenge were in multi-label format, we converted them to multi-class format by replicating the instances we used within $S$.

<table>
<thead>
<tr>
<th>Datasets</th>
<th># of classes, $K$</th>
<th>Train Size</th>
<th>Test Size</th>
<th>Heldout Size</th>
<th>Dimension, $d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSHTC1</td>
<td>12294</td>
<td>126871</td>
<td>31718</td>
<td>5000</td>
<td>409774</td>
</tr>
<tr>
<td>DMOZ</td>
<td>27875</td>
<td>381149</td>
<td>95288</td>
<td>34506</td>
<td>594158</td>
</tr>
<tr>
<td>WIKI-Small</td>
<td>36504</td>
<td>796617</td>
<td>199155</td>
<td>5000</td>
<td>380078</td>
</tr>
<tr>
<td>WIKI-50K</td>
<td>50000</td>
<td>1102754</td>
<td>276939</td>
<td>5000</td>
<td>951558</td>
</tr>
<tr>
<td>WIKI-100K</td>
<td>100000</td>
<td>2195530</td>
<td>550133</td>
<td>5000</td>
<td>1271710</td>
</tr>
</tbody>
</table>

Table 2: Characteristics of the datasets used in our experiments

**Baselines.** We compare the proposed approach\[2\] denoted as the sampling strategy by $(\pi, \kappa)$-DS, with popular baselines listed below:

- **OVA:** LibLinear \[10\] implementation of one-vs-all SVM.
- **M-SVM:** LibLinear implementation of multi-class SVM proposed in \[8\].
- **RecallTree** \[9\]: A recent tree based multi-class classifier implemented in Vowpal Wabbit.

\[2\]Source code and datasets can be found in the following repository https://github.com/bikash617/Aggressive-Sampling-for-Multi-class-to-BinaryReduction
The results are provided in terms of train and predict times, total memory usage, and predictive performance. The parameters of the datasets along with the results for compared methods are shown in Table 3.

Table 3: Comparison of the result of various baselines in terms of time, memory, accuracy, and macro F1-measure

<table>
<thead>
<tr>
<th>Data</th>
<th>OVA</th>
<th>H-SVM</th>
<th>RecallTree</th>
<th>FastXML</th>
<th>PfastReXML</th>
<th>PD-Sparse</th>
<th>(π, κ)-DS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSHTC1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 163589</td>
<td>23036s</td>
<td>48313s</td>
<td>701s</td>
<td>8564s</td>
<td>3912s</td>
<td>5160s</td>
<td>321s</td>
</tr>
<tr>
<td>d = 409774</td>
<td>328s</td>
<td>314s</td>
<td>21s</td>
<td>339s</td>
<td>164s</td>
<td>67s</td>
<td>544s</td>
</tr>
<tr>
<td>K = 12294</td>
<td>44.1%</td>
<td>36.4%</td>
<td>18.1%</td>
<td>39.3%</td>
<td>39.8%</td>
<td>45.7%</td>
<td>37.4%</td>
</tr>
<tr>
<td>MaF1</td>
<td>27.4%</td>
<td>18.8%</td>
<td>3.8%</td>
<td>21.3%</td>
<td>22.4%</td>
<td>27.7%</td>
<td>26.5%</td>
</tr>
<tr>
<td>DMOZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 510943</td>
<td>18036s</td>
<td>27355s</td>
<td>2212s</td>
<td>14334s</td>
<td>15492s</td>
<td>63386s</td>
<td>10906s</td>
</tr>
<tr>
<td>d = 9407158</td>
<td>2797s</td>
<td>3981s</td>
<td>47s</td>
<td>424s</td>
<td>505s</td>
<td>482s</td>
<td>2122s</td>
</tr>
<tr>
<td>K = 27875</td>
<td>37.7%</td>
<td>32.2%</td>
<td>16.9%</td>
<td>33.4%</td>
<td>33.7%</td>
<td>40.8%</td>
<td>27.8%</td>
</tr>
<tr>
<td>MaF1</td>
<td>22.2%</td>
<td>14.3%</td>
<td>1.7%</td>
<td>15.1%</td>
<td>15.9%</td>
<td>22.7%</td>
<td>20.5%</td>
</tr>
<tr>
<td>WIKI-Small</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 1000772</td>
<td>212438s</td>
<td>&gt;4d</td>
<td>1610s</td>
<td>10646s</td>
<td>21702s</td>
<td>16309s</td>
<td>1290s</td>
</tr>
<tr>
<td>d = 300078</td>
<td>2270s</td>
<td>NA</td>
<td>24s</td>
<td>453s</td>
<td>871s</td>
<td>382s</td>
<td>2577s</td>
</tr>
<tr>
<td>K = 36504</td>
<td>37.4%</td>
<td>32.2%</td>
<td>16.9%</td>
<td>33.4%</td>
<td>33.7%</td>
<td>40.8%</td>
<td>27.8%</td>
</tr>
<tr>
<td>MaF1</td>
<td>22.2%</td>
<td>14.3%</td>
<td>1.7%</td>
<td>15.1%</td>
<td>15.9%</td>
<td>22.7%</td>
<td>20.5%</td>
</tr>
<tr>
<td>WIKI-50K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 1384693</td>
<td>212438s</td>
<td>&gt;4d</td>
<td>1610s</td>
<td>10646s</td>
<td>21702s</td>
<td>16309s</td>
<td>1290s</td>
</tr>
<tr>
<td>d = 951558</td>
<td>2270s</td>
<td>NA</td>
<td>24s</td>
<td>453s</td>
<td>871s</td>
<td>382s</td>
<td>2577s</td>
</tr>
<tr>
<td>K = 50000</td>
<td>37.4%</td>
<td>32.2%</td>
<td>16.9%</td>
<td>33.4%</td>
<td>33.7%</td>
<td>40.8%</td>
<td>27.8%</td>
</tr>
<tr>
<td>MaF1</td>
<td>22.2%</td>
<td>14.3%</td>
<td>1.7%</td>
<td>15.1%</td>
<td>15.9%</td>
<td>22.7%</td>
<td>20.5%</td>
</tr>
<tr>
<td>WIKI-100K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m = 2750663</td>
<td>212438s</td>
<td>&gt;4d</td>
<td>1610s</td>
<td>10646s</td>
<td>21702s</td>
<td>16309s</td>
<td>1290s</td>
</tr>
<tr>
<td>d = 1271710</td>
<td>2270s</td>
<td>NA</td>
<td>24s</td>
<td>453s</td>
<td>871s</td>
<td>382s</td>
<td>2577s</td>
</tr>
<tr>
<td>K = 100000</td>
<td>37.4%</td>
<td>32.2%</td>
<td>16.9%</td>
<td>33.4%</td>
<td>33.7%</td>
<td>40.8%</td>
<td>27.8%</td>
</tr>
<tr>
<td>MaF1</td>
<td>22.2%</td>
<td>14.3%</td>
<td>1.7%</td>
<td>15.1%</td>
<td>15.9%</td>
<td>22.7%</td>
<td>20.5%</td>
</tr>
</tbody>
</table>

- **FastXML** [21]: An extreme multi-class classification method which performs partitioning in the feature space for faster prediction.
- **PfastReXML** [13]: Tree ensemble based extreme classifier for multi-class and multilabel problems.
- **PD-Sparse** [30]: A recent approach which uses multi-class loss with \( \ell_1 \)-regularization.

Referring to the work [30], we did not consider other recent methods SLEEC [5] and LEML [31] in our experiments, since they have been shown to be consistently outperformed by the above mentioned state-of-the-art approaches.

Platform and Parameters. In all of our experiments, we used a machine with an Intel Xeon 2.60GHz processor with 256 GB of RAM. Each of these methods require tuning of various hyper-parameters that influence their performance. For each methods, we tuned the hyperparameters over a heldout set and used the combination which gave best predictive performance. The list of used hyperparameters for the results we obtained are reported in the supplementary material (Appendix B).

Evaluation Measures. Different approaches are evaluated over the test sets using accuracy and the macro F1 measure (MaF), which is the harmonic average of macro precision and macro recall; higher MaF thus corresponds to better performance. As opposed to accuracy, macro F1 measure is not affected by the class imbalance problem inherent to multi-class classification, and is commonly used as a robust measure for comparing predictive performance of classification methods.

4 Results

The parameters of the datasets along with the results for compared methods are shown in Table 3. The results are provided in terms of train and predict times, total memory usage, and predictive performance measured with accuracy and macro F1-measure (MaF). For better visualization and comparison, we plot the same results as bar plots in Fig. 3 keeping only the best five methods while comparing the total runtime and memory usage. First, we observe that the tree based approaches (FastXML, PfastReXML and RecallTree) have worse predictive performance compared to the other methods. This is due to the fact that the prediction error made at the top-level of the tree cannot be corrected at lower levels, also known as cascading effect. Even though they have lower runtime and memory usage, they suffer from this side effect.

For large scale collections (WIKI-Small, WIKI-50K and WIKI-100K), the solvers with competitive predictive performance are OVA, H-SVM, PD-Sparse and (π, κ)-DS. However, standard OVA and
M-SVM have a complexity that grows linearly with \( K \) thus the total runtime and memory usage explodes on those datasets, making them impossible. For instance, on Wiki large dataset sample of 100K classes, the memory consumption of both approaches exceeds the Terabyte and they take several days to complete the training. Furthermore, on this data set and the second largest Wikipedia collection (WIKI-50K and WIKI-100K) the proposed approach is highly competitive in terms of Time, Total Memory and both performance measures comparatively to all the other approaches. These results suggest that the method least affected by long-tailed class distributions is \((\pi, \kappa)\)-DS, mainly because of two reasons: first, the sampling tends to make the training set balanced and second, the reduced binary dataset contains similar number of positive and negative examples. Hence, for the proposed approach, there is an improvement in both accuracy and \(\text{MaF}_1\) measures. The recent PD-Sparse method also enjoys a competitive predictive performance but it requires to store intermediary weight vectors during optimization which prevents it from scaling well. The PD-Sparse solver provides an option for hashing leading to fewer memory usage during training which we used in the experiments; however, the memory usage is still significantly high for large datasets and at the same time this option slows down the training process considerably. In overall, among the methods with competitive predictive performance, \((\pi, \kappa)\)-DS seems to present the best runtime and memory usage; its runtime is even competitive with most of tree-based methods, leading it to provide the best compromise among the compared methods over the three time, memory and performance measures.

5 Conclusion

We presented a new method for reducing a multiclass classification problem to binary classification. We employ similarity based feature representation for class and examples and a double sampling stochastic scheme for the reduction process. Even-though the sampling scheme shifts the distribution of classes and that the reduction of the original problem to a binary classification problem brings inter-dependency between the dyadic examples; we provide generalization error bounds suggesting that the Empirical Risk Minimization principle over the transformation of the sampled training set still remains consistent. Furthermore, the characteristics of the algorithm contribute for its excellent performance in terms of memory usage and total runtime and make the proposed approach highly suitable for large class scenario.

Acknowledgments

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d’avenir, and by the U.S. Department of Energy’s Office of Electricity as part of the DOE Grid Modernization Initiative.
References

distributions in large-scale taxonomies. SIGKDD Explorations, 16(1), 2014.
[7] Anna Choromanska and John Langford. Logarithmic time online multiclass prediction. CoRR,
[8] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector
[9] Hal Daume III, Nikos Karampatziakis, John Langford, and Paul Mineiro. Logarithmic time one-against-
[12] Eyke Hüllermeier and Johannes Fürnkranz. On minimizing the position error in label ranking. In Machine
International Conference on Knowledge Discovery and Data Mining, pages 935–944. ACM, 2016.
On binary reduction of large-scale multiclassification problems. In Advances in Intelligent Data
[17] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on
learning to rank for information retrieval. In Proceedings of SIGIR 2007 workshop on learning to rank for
[18] Ana Carolina Lorena, André C. Carvalho, and João M. Gama. A review on the combination of binary
Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto,
Amini, and P. Galinari. LSHTC: A Benchmark for Large-Scale Text Classification. ArXiv e-prints, March
2015.
[22] Liva Ralaivola and Massih-Reza Amini. Entropy-based concentration inequalities for dependent variables.
In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France,


