Sequential Dynamic Classification for Large Scale Multiclass Problems

Raphael Puget, Nicolas Baskiotis, Patrick Gallinari

Sequential Dynamic Classification for Large Scale Multiclass Problems

Raphael Puget, Nicolas Baskiotis and Patrick Gallinari

Laboratoire d'Informatique de Paris 6 (LIP6), Université Pierre et Marie Curie
Paris, France,
firstname.lastname@lip6.fr
http://www-connex.lip6.fr

Abstract. Extreme class classification concerns classification problems with very large number of classes, up to several millions. Such problems have now become quite frequent in many practical applications. Until recently, most classification methods had inference complexity at least linear in the number of classes. Several directions have been recently explored for limiting this complexity, but the challenge of learning an optimal compromise between inference complexity and classification accuracy is still largely open. We propose here a novel sequential ensemble learning approach, where classifiers are dynamically chosen among a pretrained set of classifiers and are iteratively combined in order to control efficiently the trade-off between inference complexity and classification accuracy. This model allows us to control this trade-off during the inference process which is a unique characteristic in the extreme classification literature. Experiments on a large public dataset are provided to assess the performance of the model w.r.t. a series of baselines and to analyze its behavior.

1 Introduction

Classification problems involving very large number of classes have progressively emerged over the last years and are attracting an increased attention in the machine learning community and in fields like vision, Natural Language Processing or bioinformatics. Extreme classification competitions have been recently organized, like for example the Large Scale Hierarchical Text Classification Challenge [23] with up to 320 k classes or the Imagenet Large Scale Visual Recognition challenge [4] with up to 21 k classes. Extreme classification poses several challenging issues, the first one being the control of the computational complexity, but also data scarcity and class imbalance since many classes will have only a few samples, label correlation and dependency. Also the situations and problems might be very different according to the very nature of the data itself.

Classical multi-class classification methods like one versus all have a complexity at best linear in the number of classes. Surprisingly, one versus all techniques are, up to now, among the strongest contender in term of classification performance for large class number classification problems [25, 26]. Hierarchical

methods [13, 3, 5, 18, 19, 30], relying either on existing hierarchies or on learned ones allow us to limit this complexity up to a theoretical factor (logarithmic with respect to the number of classes) which in practice is never met. Moreover, this complexity reduction most often comes with a reduction in the classification performance. These two families are prone to class imbalance and data scarcity problems. Other methods like compact class encodings may alleviate this last problem but only allow for a limited complexity reduction.

We are interested here in the development of flexible methods, with a complexity that can be adapted online to the nature and the constraints of the problem. We propose a model which, starting from a large pool of pre-trained binary classifiers, adaptively and sequentially chooses for each example during inference, an optimal ensemble of classifiers to be combined for dealing with this specific example. The sequential choice of classifiers is based on probability bounds, so that at each step the algorithm will maximize the number of potential classes that can be discarded for this example and at the same time recording informations for the relevant class. Said otherwise, the goal of the algorithm is to focus as fast as possible on a set of potential candidate classes for the example. Contrarily to hierarchical classifiers, this algorithm is able to recover from the errors of preceding classifiers during the sequential process. It also has an interesting anytime property: it can be stopped anytime and provides a guess for the class of the example.

Overall, this contribution introduces a novel way to consider large scale classification problems. By optimally choosing a sequence of classifiers among an initial pool the proposed method - named *SaDyC* for Sequential Dynamic Classification - lies between hierarchical and ensemble learning methods : the selection on the fly of the most informative classifier can be viewed as the discovery of the best hierarchy to follow for a given example; when most of classes have been discarded and a few are remaining, the behavior of our algorithm is similar to a majority vote algorithm.

Section 2 presents the related work. Our sequential decision formalism is detailed in Section 3. Experimentations on large scale datasets from LSHTC challenge and comparison with a series of state of the art baselines are presented in Section 4.

2 Related Works

One-versus-rest. As already mentioned, the basic one versus all technique remains one of the most efficient classification methods regarding classification accuracy, in the case of large class numbers. Even if it has been shown that in some contexts, it is possible to improve the performance of one versus all methods [20, 2], the simplicity and robustness of this framework [25] remains a solid choice when scalability is not an issue. It is also by nature easy to parallelize.

Hierarchical methods. Hierarchical models naturally reduce the inference time complexity. When a class taxonomy is available, one possibility is to train a classifier for each node. [20] describes a wide scope of such methods. A problem faced by hierarchical classification is that errors made on one node cannot be recovered and thus rapidly accumulate. Usually, instead of using potentially large available hierarchies, authors make use of reduced ones with only a few layers. When no class hierarchy is provided, it is still possible to learn one. Supervised clustering has recently been used in order to build such label hierarchies. Spectral clustering on interclass confusion matrices [3] for example has been proposed for learning a class hierarchy. This work has been extended to overlapping label tree [13] where the same label can be found at different nodes in the hierarchy. Such a redundancy allows a better error recovery. Other works have tried to learn a label tree by using a score function in the partitioning optimization problem [30], while some have proposed to learn a probabilistic label tree [22]. The problem of reducing the training time complexity has been explored with the conditional probability tree [5] model where the label tree is built in an online fashion, and more recently in [8].

Representation learning. Another perspective has been developed by learning compact class codes that allow a reduction in the number of classifiers. [29] and [3] proposed a way to embed the data using an existing taxonomy. The idea is to learn a class embedding that keeps similar classes close one to the other in the new space. Inference is done by computing the centroids of each class into the embedded space and by assigning the closest class for each projected example.

Ensemble methods. Ensemble methods offer an easy and natural way to control the complexity of a model by adapting the number of classifiers. Such a framework is provided by ECOC class encoding [14, 27]. [24] has proposed to improve the decoding process for ECOCs by using the loss value of each classifier during the inference process in place of the most common decoding processes (hamming, euclidean, probabilistic, laplacian). The encoding of ECOCs is usually done by randomizing the coding matrix. [9] showed an alternative and efficient way to learn an appropriate encoding in the context of large scale classification so as to improve the efficiency of each code bit. This approach provides a link between ECOCs and representation learning. Recent work on sparse coding [31] has shown how to combine efficient coding with probabilistic decoding in a large scale context.

Sequential approaches. Few works try to solve large scale multiclass classification with an anytime performance characteristic. The principle of the sequential model has been discussed in [17] which proved intuitive expected behaviors of this type of model. Sequential approaches have been used recently for dynamic feature selections like in [16] but this kind of budgeted classification is different than our budgeted context. Bandits algorithms [28, 12] have been applied in the context of on-line learning of multi-class classification problem for a small number of classes and not to optimize the inference complexity. The most related

framework of multi-armed bandit to our is recent works of [6, 7] looking at identifying the best arm rather than optimizing the trade-off exploration-exploitation but the framework is hardly transferable to our context and still main differences exist (as for instances the rewards which can not directly observed in our context, the arms can not be played twice).

3 Proposed Model

The proposed model is an iterative algorithm which accumulates sequentially positive and negative votes for subsets of classes. Given the initial set of classes $\mathcal{L} = \{\ell_0, \ell_1, \ldots, \ell_L\}$ and an example x to classify, the algorithm will output at the end of the process the class with the highest mean of positive votes as in ensemble learning methods. The algorithm 1 presents the general scheme. At each round a binary classifier is considered — named dichotomizer — which will compare two subsets of classes, C^+ and C^- ($C^+ \cap C^- = \emptyset$) and will predict if the example x belongs most likely to a class of C^+ or to a class of C^-. Positive and negative votes are recorded accordingly for the classes of C^+ and C^-. We will note in the following f_{C^+,C^-} the classifier trained to separate the set of classes C^+ from the set of classes C^- : f_{C^+,C^-} is trained such that $f_{C^+,C^-}(x) = 1$ for x from a class in C^+ and $f_{C^+,C^-}(x) = -1$ for x from a class in C^-. At each time step t, μ_i^t will denote the mean of positive votes collected for the i-th class and $\mu^t \in [0,1]^L$ the vector $(\mu_1^t, \ldots, \mu_L^t)$. As the whole set of classes is not necessary considered at each time step, we need to record the number of times that a class has been considered : the vector $T^t = (T_1^t, \ldots, T_L^t)$ will denote the number of times that each class has received a vote (and T_i^t the number of times that the class ℓ_i has received a vote).

Algorithm 1 General scheme

```
x ← example to classify
for t = 1 ... t_max do
    Select C_+ and C_− two subsets of classes of L
    Compute f_{C+,C−}(x)
    if f_{C+,C−}(x) = +1 then
        add a positive vote to classes of C+ and a negative vote to classes of C−
    else
        add a positive vote to all classes of C− and a negative vote to classes of C+
    end if
end for
return the class with highest mean of positive votes
```

The accuracy of classifiers — and thus the accuracy of the votes — is highly correlated to the number of classes in C^+ and C^- : separating two small subsets of classes is more accurate than separating two large subsets [17]. Therefore asking votes for a small subset of classes is more accurate than asking votes for

all classes. Moreover, the round is informative only if the right class belongs to one of the two subsets C^+ and C^-; otherwise the classifier will output a random vote without correlation with the class of the example to classify. The main idea of our approach is to reject at each step the most classes in order to use sparsest and thus more accurate classifiers in the next rounds to achieve a quicker identification of the right class.

This framework is close to the multi-armed bandits framework and more precisely to the best armed identification of a combinatorial multi-armed bandits problem [7]. However, instead of taking into account the upper bound confidence as in the multi-armed bandits framework, we propose to consider the confidence level that the votes of a class deviate too largely from the expected vote distribution under the hypothesis that it is the right class – which is correlated to the expected accuracy of considered classifiers. In the case that this hypothesis is wrong, the confidence will decrease throughout the iterations and eventually the class will be rejected. At the opposite, for the true class the confidence will never drop too low and the algorithm will keep collecting votes for this class. Minimizing these confidences thus answers two objectives : discarding the classes with lowest confidence while keeping at aggregating votes for classes with highest scores. The main difficulty is to select wisely at each round the subsets C^+ and C^- in order to optimize the number of discarded classes and the number of votes of the right class : this task is delegated to an oracle which will be presented in the following.

3.1 Formalization

Let $\mathcal{L} = \{\ell_0, \ell_1, \ldots, \ell_L\}$ be the set of classes defining the multi-class problem considered, with L the number of classes. For the sake of simplification, we will identify the class ℓ_i to the integer i as long as no confusion arises and $x \in \ell_i$ to denote that x is from class ℓ_i. We will use a vector $v = (v_1, \ldots, v_L) \in \{-1, 0, +1\}^L$ to encode the information of which subsets of classes C^+ and C^- are considered at each iteration : for each $\ell_i \in C^+$, v_i the i-th component of v will be set to 1; for each $\ell_i \in C^-$, the i-th component of v will be set to -1; for the other classes, the corresponding components will be set to 0. According to these notations, we will note f_v the classifier corresponding to f_{C^+,C^-} : f_v is trained such that the examples from the classes $\{\ell_i | v_i = +1\}$ are considered as positive examples and the examples from the classes $\{\ell_i | v_i = -1\}$ as negative examples; the examples of the other classes are ignored in the training phase of this classifier[1].

The norm $\|v\|$ indicates the number of classes taken into account by the classifier f_v. The term sparsity will denote the ratio $\frac{\|v\|}{L}$ for the classifier f_v. $\Phi = \{f_v | v \in \{-1, 0, +1\}^L\}$ will denote the entire set of dichotomizers for a given problem. We will suppose that a binary classifier outputs a binary decision in

[1] In the ECOC framework, such classifiers are named dichotomizers, as they separate two sets of classes.

$\{-1, 1\}$, and we will note the expected accuracy of a classifier f_v for the class ℓ_i by $q_{v,i} = \mathbb{E}_{x \in \ell_i}[v_i f_v(x) = 1]$.

3.2 Algorithm

Our algorithm aims at 1/ discarding sequentially classes over iterations and 2/ maximizing the number of votes for the right class. Our approach is based on an upper bound of the deviation of the mean μ_i of positive votes from the expectation if ℓ_i is the right label. More precisely, given an example x and its true class ℓ^*, let us notes $\mathcal{H}_i$ the hypothesis $\ell^* = \ell_i$ and X_i the random variable of the mean of positive votes for this label. After t steps where the classifiers $f_{v_1}, \ldots, f_{v_t}$ were considered, the expected mean of accuracy of the classifiers is : $Q_i^t = \mathbb{E}[X_i|\mathcal{H}_i] = \frac{1}{T_i^t}(\sum_{j=1}^{t} q_{v^j,i})$ and the observed mean is $\mu_i^t = \frac{1}{T_i^t}(\sum_{j=0}^{t} v_i^j \frac{(f_{v_j}(x)+1)}{2})$.

By using the Hoeffding inequality and noting $b_i^t = e^{\left(-2\,T_i^t\,max(0,(Q_i^t-\mu_i^t))^2\right)}$:

$$\mathbb{P}\left[\mathbb{E}[X_i|\mathcal{H}_i] - X_i \geq \delta\right] \leq e^{\left(-2\,T_i^t\,\delta^2\right)},$$

$$\mathbb{P}\left[X_i \leq \mu_i^t\right] \leq e^{\left(-2\,T_i^t\,max(0,(Q_i^t-\mu_i^t))^2\right)} \leq b_i^t.$$

These quantities will allow to discard at each round a certain number of classes by taking into account a constant b_{min} which will denote the confidence threshold. We use only one side of the Hoeffding bound, taking into account $max(0, (Q_i^t - \mu_i^t))$, as our goal is to identify when the mean of the votes is too low compared to the expectation.

The algorithm 2 resumes the principle of our approach : at each step t, a classifier f_{v^t} is considered and the quantities T_i^t, μ_i^t, Q_i^t and b_i^t are updated according to the evaluation of $f_{v^t}(x)$. The selection of the classifier is delegated to an oracle which is in charge to find the best classifier in order to minimize the confidences b_i^t, i.e. minimize $\|b^t\|_1$. The Algorithm 3 explains the mechanism of the oracle. It considers the set of dichotomizers which encode only the remaining classes $\{\ell_i | b_i^t > b_{min}\}$. For each of these dichotomizers f_v, it simulates the expected outcome if its use in the next step w.r.t. the vector b^t. The function $b^+(v)$ simulates the next state if $f_v(x) = +1$ and the function $b^-(v)$ the next state if $f_v(x) = -1$. The function $p(v)$ is used to weight the possible outcomes between $f_v(x) = +1$ and $f_v(x) = -1$ according to the current estimation.

3.3 Practical implementation and complexity

The whole set of dichotomizers Φ is generally too large to be pre-computed as it is growing exponentially with the number of classes. For the practical implementation of our model, a small sample Φ_s of classifiers is randomly chosen and pre-computed. The only control parameter taken into consideration is the number of classes encoded by the dichotomizers in order to guarantee a wide range of sparsity. In practice, for each dichotomizer to generate, a sparsity is

Algorithm 2 Inference process

x ← example to classify
Choose v_0 randomly s.t. $\|v^0\| = L$
$T_i^0 = abs(v_i^0) = (1, \ldots, 1)$
$\mu_i^0 = (f_{v^0}(x)v_i^0 + 1)/2$
$b_i^0 = \exp\left(-2t(\mu_i^0 - q_{v_i^0})^2\right)$
for $t = 1 \ldots T_{max}$ **do**
    $v_i^t = \text{Oracle}(b^{t-1}, \mu^{t-1}, T^{t-1})$
    $T_i^t = T_i^{t-1} + |v_i^t|$
    $\mu_i^t = (\mu_i^{t-1} * T_i^{t-1} + f_{v^t}(x)v_i^t)/T_i^t$
    $Q_i^t = (\sum_{j=1}^{t} q_{v_i^j})/T_i^t$
    $b_i^t = \exp\left(-2t\left(max\left(0, Q_i^t - \mu_i^t\right)\right)^2\right)$
end for
return $\arg\max_i \mu_i^{T_{max}}$

Algorithm 3 Oracle

Given b^t, μ^t, T^t, b_{min}
$\mathcal{C} = \left\{i \,|\, b_i^t > b_{min}\right\}$
define $b_i^+(v) = \exp\left(-2(t+1)(max(0, [T_i^t(Q_i^t - \mu^t) + q_{v,i} - v_i)/(T_i^t + |v_i|)])^2\right)$
define $b_i^-(v) = \exp\left(-2(t+1)(max(0, [T_i^t(Q_i^t - \mu^t) + q_{v,i} + v_i)/(T_i^t + |v_i|)])^2\right)$
define $p(v) = \frac{v \cdot b^t + \|v\|}{2\sum_{i=1}^{L} |v_i| b_i^t}$
return $\hat{v} = \arg\min_{v \in \Phi | v_i = 0 \forall i \notin \mathcal{C}} \sum_{i \in \mathcal{C}} \left(p(v)\, b_i^+(v) \;+\; (1 - p(v))\, b_i^-(v)\right)$

uniformly drawn between 0.1 to 1, a balance ratio is drawn from a Gaussian distribution centered at 0.5 to decide how many positive classes and negative classes will be considered and finally the classes are uniformly drawn among $\mathcal{L}$. The dichotomizer is learned on the training set and the expected accuracy $q_{v,i}$ of each dichotomizer is computed thanks to a validation set.

The complexity of our inference process is highly dependent on the number of dichotomizers considered and the number of steps T_{max} : the complexity τ of our model is bounded by $O((|\Phi_s| + \tau_f) \times T_{max})$ where τ_f is the inference complexity of a dichotomizer. In practice, the complexity is lower : the number of available dichotomizers is decreasing during the iterations as the number of discarded classes goes up. Moreover, in the context of large scale classification, τ_f is very large compared to $|\Phi_s|$ as it depends on the dimension of the input space — the number of features — generally more than 10^5. In comparison, one-versus-rest schema has a complexity of $O(\tau_f \times L)$, ECOC approaches $O(K \times \tau)$ with K the number of codes, and hierarchical methods $O(Qlog_Q(L) \times \tau)$ with Q the number of children per node.

We can notice that an hierarchical classification is a specific case of our approach : given a hierarchy, it is straightforward to compute a set of dichotommizers where each of them corresponds to a node of the hierarchy. Moreover, the one-versus-rest approach corresponds to the set of dichotomizers where only one

component of v is positive and all the other ones negative. Thus our approach can be viewed as a bridge between these two approaches and this fact will be analyzed in depth in the section 4.4.

4 Experiments

4.1 Datasets

The experiments were done using real world data publicly available. The first dataset comes from the recent series of challenges named LSHTC [21]. The dataset used (called DMOZ, or also know as ODP — Open Mozilla Directory) represents text documents that can be classified over 12,294 different classes organized in a class hierarchy. The dataset is already preprocessed with a TFIDF transformation directly applied to the word count. The test and validation sets used are the same as the ones from the challenge.

In order to evaluate the scalability of our method, we compared the different models on the whole DMOZ dataset and on smaller datasets with 1000 classes. The 1000 classes datasets have been extracted from the DMOZ datasets by sub sampling classes from the 12,294 original classes. More specifically, we randomly sampled one class and we computed the 999 nearest classes (with cosine norm based distance between classes censtroïds). Five such datasets were generated. Sector [**?**] is another text classification dataset with 105 classes. These different datasets with several orders of magnitude in the number of classes are used to show the scaling capacity of our approach. Statistics of the different datasets are summed up in table 1.

Table 1: Statistics of the datasets used.

	DMOZ (full)	DMOZ (sub sampled)	Sector
# training instances	93805	~9400	6992
# validation instances	34905	~3500	1469
# test instances	34880	~3500	1158
# features	347255	347255	55198
# classes	12294	1000	105

4.2 Protocol

We compared our *sequential* model with state of the art hierarchical and flat methods allowing a speed-up of the inference process. We assessed the performances of our method by comparing it with label tree partitioning using spectral clustering as described in [3] and with ECOC methods [15, 1].

We also compare the results obtained by the one-versus-rest method and by learning a label tree over the original hierarchy for DMOZ dataset - as this information is available only for this dataset.

In order to compare at different speed-up the performance of the models, we recorded the accuracy of each model w.r.t the complexity ratio gain : it represents the number of classifiers used by each model normalized by the number of classes. Thus, the complexity ratio will always be 1 for one-versus-rest method by definition. For methods which achieved a speed-up, it will be inferior to one. The lower the complexity ratio, the better the speed-up is.

For the ECOC implementation, we randomized five thousands ternary coding matrices as proposed in [1] and we kept the one that maximizes the hamming distance between each code of each class. Then, we decoded the input following two decoding schemes: the classic hamming distance [15] and the more elaborated loss-based decoding also showed in [1]. The different speed-up were obtained by modifying the length of the ECOC codes.

We controlled the speed-up of the spectral clustering method by performing experiments with different numbers of children per node in the inferred hierarchy and we reported the best results obtained.

For our model, we used a pool of approximately ten thousands randomly generated classifiers for the DMOZ dataset and 300 for the Sector dataset. The complexity ratio gain is controlled by varying the maximal number of iterations allowed. We fixed the minimum confidence constant b_{min} to 0.01.

In all the methods, the classifiers used (in the node, or for the ECOC bit scorer) were SVM [10] regularized over a validation set[2].

4.3 Results

Fig. 1 shows the results of the compared methods on the whole DMOZ dataset. The one-versus-rest approach (label *OAA*) has the best accuracy but for the worst complexity ratio, equal to 1. The hierarchical label tree learned from the original hierarchy (label *H-Ontology*) has an accuracy close to the one-versus-rest for low complexity ratio, but this approach is the only one among the studied methods who uses *a priori* information on classes to build its model.

The hierarchy learned from spectral clustering (label *H-SC*) shows impressive performance for very low complexity ratio (under 1%) but its accuracy is rapidly bounded and does not benefit from more allowed resources. At the opposite, ECOC models (label *E-Dense* for hamming distance decoding and *E-LossBased* for loss-based decoding) show medium performances for very low complexity ratio but the accuracy is growing fast when more classifiers are used. Our proposed model (label *SaDyC*) has an accuracy smaller but close to the hierarchical clustering for very low complexity ratio ($< 1\%$) and has the overall best performances when the complexity ratio goes up among methods that do not use *a priori* information.

Fig. 3 reports the results for the sampled datasets with 1000 classes from DMOZ. Similar conclusions as previously can be drawn for the compared approaches. Moreover, when enough resources are allowed (ratio bigger than 10%),

[2] We used the implementation from *scikit learn* library (`LinearSVC`) on big datasets (only for linear SVM). The same meta parameters were used in all the compared methods (*l2* loss, automatic class weighting and *l2* penalty).

our model is able to challenge and in some cases to outperform the *H-Ontology* model that uses *a priori* information. Interestingly when complexity ratio is bigger than 15% our approach has a better accuracy than the one-versus-rest algorithm.

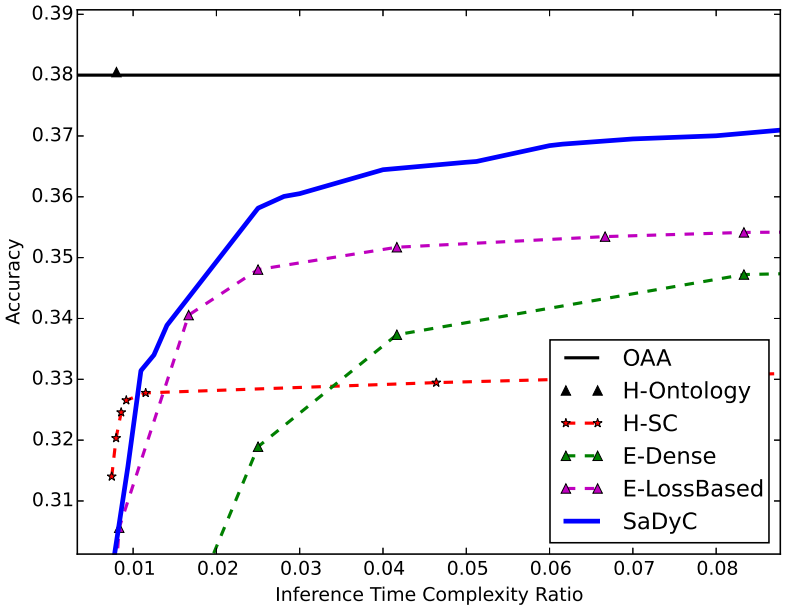

Fig. 1: Accuracy of compared methods on full DMOZ dataset for different complexity ratio values.

Tables 2 and 3 report the accuracy results of the compared methods for a given complexity ratio of 4% on the DMOZ dataset and subsample datasets of 1000 classes. It confirms the observations that we reported previously.

Table 4 shows the results for Sector dataset with only 100 classes for a complexity ratio of 20% : similar results are observed than with the large DMOZ dataset, our algorithm *SaDyC* outperforming the other approaches[3].

4.4 Discussion

One interesting point to discuss is the ability of the hierarchical methods to perform well for very extreme speed-up. This can be explained by the fact that

[3] We did not report directly in the table the result of [8] on this dataset as they did not used the same dichotomizer type but it can be nonetheless noted that our model performed better with a large margin.

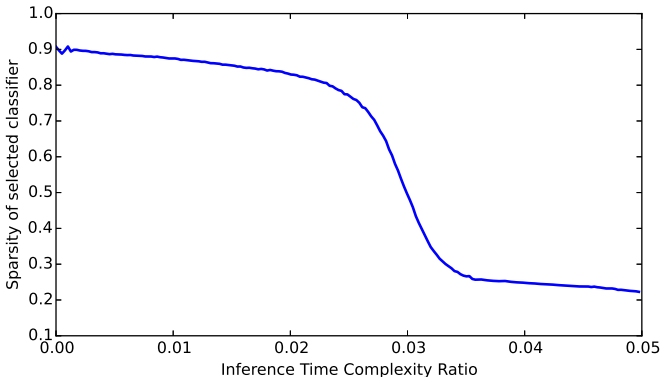

Fig. 2: Density (1 - Sparsity) of selected classifiers during inference process on the full DMOZ dataset.

hierarchical structure allows easily large speed-up. The *fast* hierarchy is able to classify well the most easily classified examples with less errors than other methods. As more computation resources are allowed, the hierarchical spectral clustering based model lost its advantage due to the fact that it is not purposely designed to offer anytime performances.

What is also interesting with these results is the comparison of our model with the ECOC Loss-based model. At the greater speed-up, both perform similarly. As the inference computation time constraint is relaxed, our sequential approach shows the best performances as it is designed to use efficiently the pool of classifiers that was given to it. This comes from the ability of our approach to discard bad classes in order to focus on more valuable information by choosing accordingly the next classifier to use. We define classifier density the density of the code corresponding to a given classifier. A density of 1 means that every classes are taken into account in the classifier. Fig. 2 shows the decrease of the density of the classifiers used through the iterations of our algorithm. ECOC models did not propose such ability. They are bound to use the same classifiers for each new example. To understand this ability of our model, another point of view is to see our approach as a hierarchical one : for a given example, the trajectory of the selected classifiers is similar to a path along a hierarchy. But rather than considering always the same hierarchy as the usual hierarchical approach, our model is able to discover the most promising one in the first steps of the algorithm. The Fig. 2 shows that *SaDyC* spends many iterations considering not sparse classifiers, where the hierarchical approach can consider only for a small number of iterations large number of classes, even if redundancy is allowed in the label tree (i.e. same labels can be in more than one child). Once the right hierarchy is identified, the sparsity fall quickly until there is few classes to separate. At this stage, our approach is able to use at best the benefit of ensemble learning methods in order to improve its accuracy.

It can be observed that one-versus-rest model performs better relatively to all the other methods on the big DMOZ dataset than on the sub sampled datasets. This is explained by the kind of sampling that has been done to produce the sub sampled datasets. With less classes to separate, the dichotomizers of the other methods were more accurate than for the big DMOZ dataset.

Table 2: Accuracy results for 13K classes on full DMOZ dataset. We reported in bold font the best significant results. We put in italic the results of models that can not be compared directly with the other methods as they use more information (H-Ontology) or more inference time (OAA). Same fonts were used in the next tables.

Models	Ensemble Type	Complexity Ratio	DMOZ (full) Acc%
One-vs-Rest (OAA)	Flat	1 (×1)	***38.28***%
Hierarchical Ontology (H-Ontology)	Tree	0.008 (×125)	*38.05*%
SaDyC	Flat	0.04 (×25)	**36.49**%
Hierarchical Spectral Clustering (H-SC)	Tree	0.04 (×25)	32.09%
ECOC *Hamming* (E-Hamming)	Flat	0.04 (×25)	33.68%
ECOC *Loss Based* (E-LossBased)	Flat	0.04 (×25)	35.08%

Models	Ensemble Type	Complexity Ratio	DMOZ (sub-sampled): Acc% Set1	Set2	Set3	Set4	Set5
OAA	Flat	1 (×1)	*45.50*%	*55.36*%	*60.83*%	*54.86*%	*52.16*%
H-Ontology	Tree	0.06 (×16)	*46.73*%	*57.15*%	***63.16***%	***57.08***%	*54.84*%
SaDyC	Flat	**0.2** (×5)	**49.78**%	**57.43**%	**62.36**%	**57.05**%	**55.29**%
H-SC	Tree	0.2 (×5)	44.77%	53.72%	58.29%	52.63%	51.97%
E-Hamming	Flat	0.2 (×5)	45.03%	51.99%	56.98%	52.38%	50.89%
E-LossBased	Flat	0.2 (×5)	46.72%	54.42%	59.15%	54.20%	52.19%

Table 3: Accuracy results for 1K classes sub-sampled DMOZ datasets.

Besides the overall good performances of the proposed method, a key feature is that our model is the only one able to stop the inference process on the fly while ensuring top performances. This anytime characteristic has, up to our knowledge, not been studied in the literature for the large scale classification problems.

5 Conclusion

We presented in this paper a novel approach to deal with large scale multi-class classification tasks. Our sequential model can produce an accurate answer

Models	Ensemble Type	Complexity Ratio	Sector
OAA	Flat	1 (×1)	***94.12***%
SaDyC	Flat	0.2 (×5)	**92.67**%
H-SC	Tree	0.2 (×5)	88.52%
E-Hamming	Flat	0.2 (×5)	27.73%
E-LossBased	Flat	0.2 (×5)	89.55%

Table 4: Accuracy results for the Sector dataset.

with an anytime performance characteristic that has many possible applications nowadays. From a pool of classifiers, the proposed model uses an oracle to select at each time step the most accurate classifier in order to optimally discards classes to keep only the ones of interest and at the same time recording more informations on classes with high probabilities to be the targeted ones. Thus, the algorithm can use more specific classifiers throughout the iterations. The proposed approach can be viewed as a hierarchical one where there is no specific hierarchy at the beginning of the process and first steps are used to discover the most promising one. Our experiments show how our model performs better than state of the art methods for similar speed-up factors.

The focus of this paper was on how to use a large pool of classifiers the most effectively. The actual tuning of the pool of classifiers is a whole different problematic. As shown in [11], the accuracy of the classifiers used in the inference process impacts the theoretical maximum accuracy bound of the overall classification process. Thus, the learning of an adequate pool of classifiers has a lot of potential to greatly improve the performances of the actual presented model.

Another perspective is to control finely in an online fashion the compromise between classification accuracy and execution time. It will allow the model to adapt to application constraints. For instance in the case of an online classification task on a data stream, the latter can fluctuate so that when the stream speeds up, the allowed computation time may be reduced and when the stream slows down, computation time may be increased.

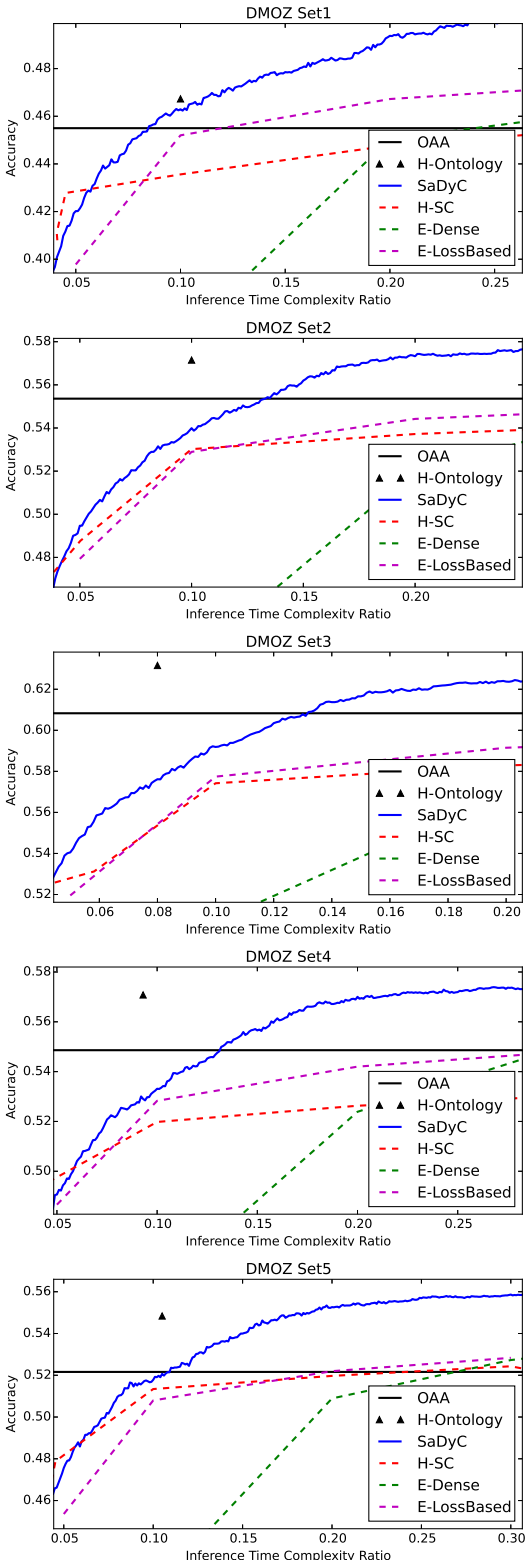

Fig. 3: Accuracy results on 5 sub sampled datasets of DMOZ for compared methods.

References

1. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. J. Mach. Learn. 1, 113–141 (2001), http://dl.acm.org/citation.cfm?id=944737
2. Babbar, R., Partalas, I.: On Flat versus Hierarchical Classification in Large-Scale Taxonomies. Neural Inf. Process. Syst. pp. 1–9 (2013), http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2013_5082.pdf
3. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. Adv. Neural Inf. Process. Syst. 23(1), 163–171 (2010), http://0-research.google.com.topcat.switchinc.org/pubs/archive/36578.pdf
4. Berg, A., Deng, J.: Imagenet large scale visual recognition challenge 2010. Challenge (2010), http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Large+Scale+Visual+Recognition+Challenge+2010\#2
5. Beygelzimer, A., Langford, J., Lifshits, Y., Sorkin, G., Strehl, A.: Conditional Probability Tree Estimation Analysis and Algorithms. Uncertain. Artif. Intell. (2009)
6. Bubeck, S., Munos, R., Stoltz, G.: Pure exploration in multi-armed bandits problems. Algorithmic Learn. Theory (2009), http://link.springer.com/chapter/10.1007/978-3-642-04414-4_7
7. Chen, S., Lin, T., King, I., Lyu, M., Chen, W.: Combinatorial Pure Exploration of Multi-Armed Bandits. Neural Inf. Process. Syst. pp. 1–9 (2014), http://papers.nips.cc/paper/5433-combinatorial-pure-exploration-of-multi-armed-bandits
8. Choromanska, A., Langford, J.: Logarithmic Time Online Multiclass prediction pp. 1–13 (2014)
9. Cissé, M., Artières, T., Gallinari, P.: Learning compact class codes for fast inference in large multi class classification. Eur. Conf. Mach. Learn. (2012), http://link.springer.com/chapter/10.1007/978-3-642-33460-3_38
10. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20, 273–297 (1995)
11. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Mach. Learn. pp. 201–233 (1995), http://link.springer.com/article/10.1023/A:1013637720281
12. Crammer, K., Gentile, C.: Multiclass classification with bandit feedback using adaptive regularization. Mach. Learn. 90, 347–383 (2013)
13. Deng, J., Satheesh, S., Berg, A., Fei-Fei, L.: Fast and Balanced: Efficient Label Tree Learning for Large Scale Object Recognition. In: Neural Inf. Process. Syst. pp. 1–9. No. 1 (2011), http://books.nips.cc/papers/files/nips24/NIPS2011_0391.pdf
14. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. arXiv Prepr. cs/9501101 2 (1995), http://arxiv.org/abs/cs/9501101
15. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. (1995), http://arxiv.org/abs/cs/9501101
16. Dulac-arnold, G., Denoyer, L., Preux, P., Gallinari, P.: Datum-wise classification. A sequential Approach to sparsity. ECML/PKDD pp. 375–390 (2011), http://eprints.pascal-network.org/archive/00008290/
17. Even-Zohar, Y., Roth, D.: A sequential model for multi-class classification. EMNLP (2001), http://arxiv.org/abs/cs/0106044
18. Gao, T., Koller, D.: Discriminative learning of relaxed hierarchy for large-scale visual recognition. ICCV (2011), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6126481

19. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. Comput. Vis. Pattern Recognit. pp. 1–8 (Jun 2008), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587410
20. Jr, C.S., Freitas, A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Discov. 44(0) (2011), http://www.jstor.org/stable/2981629http://link.springer.com/article/10.1007/s10618-010-0175-9
21. Kosmopoulos, a., Gaussier, E., Paliouras, G., Aseervatham, S.: The ECIR 2010 large scale hierarchical classification workshop. ACM SIGIR Forum 44, 23 (2010)
22. Liu, B., Sadeghi, F., Tappen, M., Shamir, O., Liu, C.: Probabilistic label trees for efficient large scale image classification. Comput. Vis. Pattern Recognit. pp. 843–850 (2013)
23. Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.R., Gallinari, P.: LSHTC: A Benchmark for Large-Scale Text Classification (2015)
24. Passerini, A., Pontil, M., Frasconi, P.: New results on error correcting output codes of kernel machines. IEEE Trans. neural networks 15(1), 45–54 (2004), http://www.ncbi.nlm.nih.gov/pubmed/15387246
25. Perronnin, F., Akata, Z., Harchaoui, Z., Schmid, C.: Towards good practice in large-scale learning for image classification. 2012 IEEE Conf. Comput. Vis. Pattern Recognit. pp. 3482–3489 (2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6248090
26. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. 5, 101–141 (2004), http://dl.acm.org/citation.cfm?id=1005336
27. Schapire, R.: Using output codes to boost multiclass learning problems. ICML (1), 1–9 (1997), http://www.cs.iastate.edu/~jtian/cs573/Papers/Schapire-ICML-97.pdf
28. Wang, S., Jin, R., Valizadegan, H.: A potential-based framework for online multi-class learning with partial feedback. J. Mach. Learn. Res. 9, 900–907 (2010), http://www.scopus.com/inward/record.url?eid=2-s2.0-84862291188\&partnerID=tZOtx3y1
29. Weinberger, K., Chapelle, O.: Large margin taxonomy embedding with an application to document categorization. Adv. Neural Inf. pp. 1–8 (2008), http://www.cse.wustl.edu/~kilian/papers/taxo_nips.pdf
30. Weston, J., Makadia, A., Yee, H.: Label partitioning for sublinear ranking. Int. Conf. Mach. Learn. 28 (2013), http://machinelearning.wustl.edu/mlpapers/papers/weston13
31. Zhao, B., Xing, E.P.: Sparse output coding for large-scale visual recognition. Comput. Vis. Pattern Recognit. pp. 3350–3357 (2013)