Scalable Learnability Measure for Hierarchical Learning in Large Scale Multi-Class Classification

Raphael Puget, Nicolas Baskiotis, Patrick Gallinari

Scalable Learnability Measure for Hierarchical Learning in Large Scale Multi-Class Classification

Raphael Puget, Nicolas Baskiotis, Patrick Gallinari
Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie, Paris, France.
firstname.lastname@lip6.fr

ABSTRACT

The increase in computational and storage capacities leads to an increasing complexity of the data to be treated: data can be represented in much more detail (many features) and in very large amounts : in the context of text categorization or image classification, the number of labels can scale from 10^2 to 10^5, and features range from 10^4 to 10^6. The main trade-off is generally between the accuracy of the predictions and the inference time. A usual methodology consists in organizing multiple classifiers in a hierarchical structure in order to reduce the computation cost of the inference. A popular category of algorithms is to iteratively build the structure. Inspired by clustering, the iteration scheme is a splitting (top-down algorithms) or aggregating (bottom-up algorithms) process. This step uses measures to determine the split/aggregation rule (like entropy, similarity between classes, separability ...). These kinds of measures are often computationaly heavy and can not be used in a large scale context. In this paper, we propose to use a reduced projected space of the input space to build measures of interest. Preliminary experiments on real dataset show the interest of such methods. We propose preliminary experiments which integrate a "learnability" measure in hierarchical approaches.

1. INTRODUCTION

In the context of large scale multi-class classification, the speed off of the inference time is becoming a real bottleneck for the industrial applications. Moreover, the learning process is quite heavy, computationally intensive, and thus lighter methods are required to achieve practical use.

Approaches usually rely on reducing the number of classifiers. The trade-off is between the accuracy obtained and the inference time under the condition of a reasonable learning time.

Two main approaches exist for this task : reducing the label space by combining classifier decisions (mainly ECOC [8, 5]) and organizing classifiers in a graph or tree structure to speed up decision process [4]. In this latter setting, each node of the structure represents itself a learning subproblem consisting in predicting the right successor of the current node for a data. Mainly three nested optimization problems occurred : optimizing the architecture of the structure (how deep, how many children by node, ...); optimizing the affectation of learning problems to the structure, i.e. find a well-adapted labels partition to be mapped into the structure; and optimizing the classifiers at each node globally.

When an a priori knowledge is available on labels, it can be used to search the more appropriate hierarchy or combination of classifiers [14]. However, there is no guarantees that the label similarity can help at the learning phase; furthermore, such hierarchies are often flat, with many branches per node, do not respect data balancing, and are often unavailable. Building a hierarchy from a large collection of label without any complementary information (ontologies, semantic similarities) without decreasing severely the accuracy of the prediction is a real challenge. Due to the large degree of freedom (in terms of nodes degree, structure, model selection, ...), global optimization is often a very hard problem.

Many heuristics have been studied extensively for the construction of hierarchies for learning : bottom up approaches [10]; top down approaches [7, 4, 6]; online approaches [3]; global optimization problem [9]. A fundamental aspect of the large scale classification task is that in a general way, most naive approaches as the One-versus-Rest performed the best in term of accuracy [12]. The challenge is still to find methods that can reach the most expensive options.

In this paper we focus on hierarchical bottom-up strategies based on a learnability criterion. We introduce an estimation measure of the learnability of two subsets based on dimension reduction. We propose in the following to study an ECOC-like projection to speed-up the criterion computation by largely reducing the number of dimension and to improve the accuracy of the computed information. A learnability measure is next proposed which use this projection to assess the pertinence of our approach. Preliminary empirical results show that our strategy outperforms baseline aggregating top-down strategies and usual ECOCs.

The paper is organized as follow : Section 2 presents notations and definitions; section 3 discuss general considerations on tree architecture for large scale classification. Section 4

presents our algorithm and experiments on real datasets.

2. NOTATIONS AND DEFINITIONS

We consider in the following classifiers organized in a hierarchical way according to a tree, i.e. without cycles. An example is classified as in classical decision tree, by following from the root a sequence of tests at each node which indicates the next node of the classification path. The process ends when a leaf is reached and the example is classified according to the leaf (generally a majority vote is used).

A learning problem is defined by Y a set of labels, a training set $X \subset \mathbb{R}^d \times Y = \{(x_i, y_i)\}$, with y_i the label associated to the example x_i. We choose to formalize the decision function at each node by a set of classifiers with a probabilistic output, one per child, denoting the probability that an example belong to the associated children.

A such classification tree is noted $T = (N, E, F, L)$, with N the set of nodes indexed by $\{1, \ldots, n\}$, E the set of edges, $F = \{f_{(parent,children),\ldots}\}$ the set of decision functions associated at each edge, and L the label sets associated to each node.

The classification path of an example x is the sequence of nodes that leads to the leaf corresponding to x : $path(x) = \{i_1, \ldots, i_l\}$ s.t i_1 is the root and $i_{k+1} = argmax_{j \in ch(i_k)} f_{i_k,j}(x)$, $ch(i)$ the set of children of node i.

We will focus on the zero-one loss in the following. We note $f_T : R^d \rightarrow Y$ the decision function associated to the tree T. The empirical loss is : $R(f_T) = \frac{1}{|X|} \sum_X I(f_T(x_i) \neq y_i)$.

3. AGGREGATING ALGORITHM

3.1 Speeding up with tree architecture

In order to compare easily the speed-up of our methods, we define the compression classification time ratio r that corresponds to the time needed to infer an example with the inferred hierarchical model over the time needed for a flat one-versus-rest SVM[1]. If r is equal to 1, the inference time is similar to a flat method. Inference is more rapid for smaller values of r.

Our approach is based on the main heuristic that without any accurate fitness measure on the quality of the partitioning (indicating the learnability of partition problems at a node), the best generic structure for a ratio r with bounded resources $2^{|Y|}$ classifiers is a first level with a non-fixed number of children, while all the other subtrees are binaries (thus all nodes excepting the root have 2 children). From a time cost point of view, adding branches to deep node has poor effect, as the impact of the modifications decreases exponentially with respect to the depth. On the other hand, flat one-versus-rest structure is known to be among the best classification pattern [12] (very recent researches [1] show post-processing hierarchies methods getting better results than flat models). Moreover, the impact of the errors decreases exponentially with the distance to the root. Thus, to minimize the loss without prior knowledge, the best architecture consists in focusing all the discriminative power at the root of the tree. These two considerations argue for the proposed structure, with balanced subtrees complexity at the first level.

Experiments have been conducted to study empirically the influence of tree basic structure on the LSHTC dataset[2]. Two kind of structures have been considered : a binary first level and the subtrees are k-ary balanced trees (with same structure), named structure A; a root with k-children and each child is a binary tree, named structure B. These two structures A and B corresponds to tree skeletons. The labels are then affected randomly to the nodes of these two structures. Over the 10,000 random experiments, for the same compression time ratio, the structure B showed better scores than structure A with a difference of 3 to 5 points for different ratio r targeted.

3.2 Bottom-up algorithm

The general algorithm of a bottom-up agglomerative approach consists in maintaining a forest of trees and to choose iteratively 2 trees to merge based on a given measure, i.e. create a new node which will be the parent of the two selected tree roots, until the forest contains only one tree. At each step, the process can be stopped and a tree can be obtained by grouping all the trees of the forest under a same root node.

Algorithm 1 Bottom up agglomerative strategy

1: **Init.** : $Forest = \{T_1, \ldots, T_{|Y|}\}$ the set of 0-depth trees with one unique node, each one tagged by label l_i, which will be the leafs set of the final tree.
2: **while** Complexity($Forest$) $<$ targeted complexity **do**
3: Find two trees T_i, T_j which minimize a criterion
4: Create a new node i', remove T_i and T_j from $Forest$ and add the tree with root i' and children T_i, T_j. Learn the classifiers $f_{i',i}$ and $f_{i',j}$ corresponding to the two new edges.
5: **end while**
6: **Merge** all the remaining trees together (under the root node) and learn the associated classifiers.

A widely used measure to perform the tree pairing is based on a confusion matrix computed from a set of linear one-versus-rest SVM. The final measure we used is a trade-off between the confusion matrix information and the size difference (in term of number of training examples) of the two trees.

4. SCALABLE LEARNABILITY MEASURE

The criterion to pair at each step two trees requires to be fast to compute in very large scale context. [11] has shown that current methods are not sufficiently accurate nor scalable to perform good classes pairing. We propose in the following to study an ECOC-like projection to speed-up the criterion computation by largely reducing the number of dimension and to improve the accuracy of the computed information. A learnability measure is next proposed which use this projection to assess the pertinence of our approach.

[1]The time complexity in the following is the number of classifiers used to predict the class of an example. In the case of the one-versus-rest, the time complexity is equal to the number of classes.

[2]http://lshtc.iit.demokritos.gr/

4.1 ECOC projection

In order to tackle the problem of scalability, we propose to estimate measures in a space with many less dimensions than the original one. We use for that an ECOC-like projection. Given a random vector $v = \{-1, 0, 1\}^{|Y|}$ which affect -1, 0 or 1 to each label of the learning set (uniformly according to a ratio of sparsity), a binary classifier with probabilistic output $\phi_v : X \to [0, 1]$ is learnt such that examples from labels tagged by -1 (resp. 1) are considered negatives (resp. positives) and the 0 examples are ignored. The new feature for an example $x \in \mathbb{R}^d$ is the output of $\phi_v(x)$. Considering m random vectors, the projection of an example x is the vector $[\phi_{v_1}(x), \ldots, \phi_{v_m}(x)]$. This projection allows to produce random projected space but the projection is guided by the labels.

4.2 Learnability measure

Recent studies pointed out theoretically that sparse separability is highly related to the complexity of boosting trees [13]. We propose to use the overfitting capabilities of boosting trees and their computational effectiveness in large scale context to regroup classification problems by increasing difficulty. We consider that given two comparable learning problem settings, the easiest one is the one which requires the less boosting epochs to overfit.

In order to integrate our criterion, we propose the following modification of the bottom up algorithm: at each step, a filter is first apply in order to select trees of the forest such that each binary classification problem resulting from any two merge of the roots have the same low difficulty among all the possible merge. After that, the measure presented in 3.2 is used to select the two trees to be merged.

In practice, we consider $\{B_1, \ldots, B_b\}$ a set of classifiers family, such that the complexity and the expressiveness of B_i is increasing. An example of such family is boosting trees where each B_i corresponds to an upper bounded allowed epochs (or number of trees considered), increasing with i. Set the boolean predicate $learn_k(i, j)$ between two nodes i, j s.t. it is true if the binary learning problem which consists in discriminating examples from node i and node j is overfitted by B_k. If $learn_k(i, j)$ is true, then for $k' > k$ $learn_{k'}(i, j)$ is true. The filtering step consists in finding the maximal clique for this relation s.t. k is minimal and the set of candidates to merge is not empty.

4.3 Experiments/analysis

We tested our method on real data that comes from the challenge LSHTC. We made sub-datasets of 100 classes of the original one that contains more than 12,000 classes and 300,000 features. In order to keep challenging datasets, the 100 classes of the sub-datasets were picked by selecting classes close from each other.

Each set is composed of approximately 10,000 examples which are then decomposed into a train, test and validation set ($\frac{3}{5}$, $\frac{1}{5}$ and $\frac{1}{5}$). The mean number of features is 50,000. The parameters guiding the partitioning were tuned by using the validation set.

We assessed two methods of our own. One is a naive version of our *High Complexity Structure Tree* with naive bottom-up agglomerative algorithms (HCST Greedy). The other one use complexity family to guide the merges (HCST Complexity). We compare our methods to a naive classical ECOC method, to a top-down partitioning based on spectral clustering on a confusion matrix (Bengio's partitioning [2]) and finally, to a One-Versus-Rest linear SVM.

For all the tree partitioning methods, a linear SVM is trained at the end for each node of the tree hierarchy.

For the ECOC projection, we use a code of length 20 (to compare with the original space size of 50000 features), a sparsity of 1/3 and balanced distribution of 1 and -1. For the set of classifiers family $\{B_1, \ldots, B_b\}$, we consider boosting trees with the number of trees limited by the series $\{1, 2, 4, 8, 16\}$.

The classification accuracy computed is the mean of per-class accuracy. To assess the computational gain, we computed the mean of the number of classifier evaluations for all train instances. Then, we compared all these methods for a fixed computational gain. This computational gain is expressed as a ratio between 0 and 1 called *Complexity Ratio*. It is the ratio of the mean of the number of classifier used for inference over the number of classes. Though, the complexity ratio for the One-Versus-Rest method is 1 and every ratio below 1 has a quicker inference time than One-Versus-Rest.

As expected, One-vs-Rest model achieves the best performances. The naive baseline (HCST Greedy) underperforms generally One versus Rest by 2 to 3 points (except on set 3). Our approach ranges between these two boundaries, and in some sets achieving very close to the One-vs-Rest. The ECOC and top-down partitioning are beaten in every experiments, but these two algorithms were used in their naive version without any tunning of the parameters.

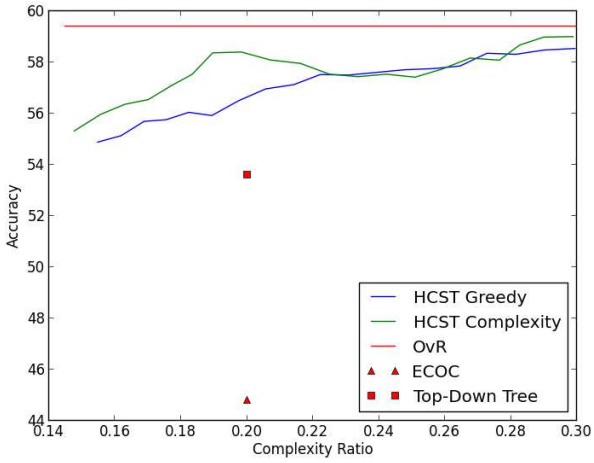

Figure 2: Accuracy with respect to Complexity Ratio for Set4 experiment

These preliminary results show first of all that a measure of learnability can be used successfully in the large scale classification setup by estimating a such criterion on a reduced dimensional space. It appears more stable than other sim-

Classifier	Ensemble Type	Computational Ratio	LSHTC: Acc% (std) Set1	Set2	Set3	Set4
One-vs-Rest	Flat	1	60.5% (0.2)	75.05% (0.6)	81.9% (0.2)	59.4% (0.5)
HCST Complexity	Tree	**0.2**	**58.75**% (0.3)	**74.56**% (0.4)	81.34% (0.1)	**58.7**% (0.6)
HCST Greedy	Tree	0.2	58.12% (0.5)	73.3% (0.7)	**81.5**% (0.1)	56.6% (0.4)
Top-Down Tree	Tree	0.196 | 0.194 | 0.202 |0.198	54.47% (1.3)	71.65% (0.8)	79.47% (0.6)	53.6% (1.1)
ECOC	Flat	0.2	47.07% (3.0)	61.38% (8.6)	73.1% (1.7)	44.8% (3.0)
ECOC	Flat	0.5	57.27% (0.9)	72.93% (0.9)	80.6% (0.7)	56.1% (1.1)

Figure 1: Flat versus Tree Results

ilarity measures as confusion matrices. One explanation is that the space reduction allows to denoise globally the partitions and thus the information extracted is more stable than in the original space. On the other hand, few tries on a datasets of 1000 classes shows no difference between the HCST greedy and HCST complexity. One could expects this phenomenon as the dimensionality reduction tends to produce more uniform features as the merging process progress : the encoding is not recomputed and the dispatch between positive and negative classes for the codeword is fixed at the beginning of the process; at the end of the merging process, positive and negative classes are generally uniformly distributed in the different partitions, and thus loose their compression competence. Moreover, our detection of overfitting is very naive and does not adapt well under different learning settings. These preliminary results show however that learnability criterion can achieve good performances and projection of input space can be used not only to learn or clusterise, but also to estimate in a very efficient way properties of the input space.

5. CONCLUSION AND PERSPECTIVES

In this work we show experimentally the interest to use a learnability measures to improve the accuracy in a particular setting of many classes classification. More interesting, the experiments shows that the criterion can be estimated in a projection space through dimensionality reduction and thus be scalable. It is particularly important in the context of exploration strategies for large scale classification : the context involves many simulations and estimations to deal with the number of partitions available and estimating accurate fitness measure quickly and scalable is a real bottleneck to many approaches. Further studies are needed to explore intensively the exact role and use of the dimensionality reduction. We want to extend this work in two main ways : considering recoding schema in order to tackle the problem of the increasing inaccuracy of the measure in the a priori fixed projected space; adapting this kind of measures to assess quality of partition with respect to others. The long-term goal is to design top-down or bottom-up strategies adaptable to the decomposition of the combinatorial problem aspect of large scale classification.

Acknowledgments. This work was supported in part by the ANR project Class-Y and the European project BioASQ (grant agreement no. 318652).

6. REFERENCES

[1] Rohit Babbar and Ioannis Partalas. On Flat versus Hierarchical Classification in Large-Scale Taxonomies. *Neural Information Processsing Systems (NIPS)*, pages 1–9, 2013.

[2] Samy Bengio, J Weston, and D. Grangier. Label embedding trees for large multi-class tasks. *Advances in Neural Information Processing Systems*, 23(1):163–171, 2010.

[3] Alina Beygelzimer, John Langford, Yuri Lifshits, Gregory Sorkin, and Alex Strehl. Conditional Probability Tree Estimation Analysis and Algorithms. *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

[4] W Bi and J Kwok. Multi-label classification on tree-and DAG-structured hierarchies. *Yeast*, pages 1–8, 2011.

[5] M Cissé, T Artières, and P Gallinari. Learning compact class codes for fast inference in large multi class classification. *European Conference on Machine Learning (ECML)*, 2012.

[6] C Cortes and V Vapnik. Support-vector networks. *Machine learning*, 1995.

[7] Jia Deng, Sanjeev Satheesh, A. Berg, and L. Fei-Fei. Fast and Balanced: Efficient Label Tree Learning for Large Scale Object Recognition. In *NIPS*, number 1, pages 1–9. NIPS, 2011.

[8] TG Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Arxiv preprint cs/9501101*, 1995.

[9] Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. *ICCV*, 2011.

[10] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[11] M Marszalek and Cordelia Schmid. Constructing category hierarchies for visual recognition. *Computer Vision - ECCV*, 2008.

[12] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3482–3489, June 2012.

[13] S Shalev-Shwartz and Yoram Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. *Machine learning*, 2010.

[14] Kilian Weinberger and Olivier Chapelle. Large margin taxonomy embedding with an application to document categorization. *Advances in Neural Information*, pages 1–8, 2008.