On Empirical Tradeoffs in Large Scale Hierarchical Classification

Rohit Babbar, Ioannis Partalas, Éric Gaussier, Cécile Amblard

On Empirical Tradeoffs in Large Scale Hierarchical Classification

Rohit Babbar, Ioannis Partalas, Eric Gaussier, and Cecile Amblard
LIG, Université Joseph Fourier, Grenoble 1
Grenoble, cedex 9, France, 38041
firstname.lastname@imag.fr

ABSTRACT

While multi-class categorization of documents has been of research interest for over a decade, relatively fewer approaches have been proposed for large scale taxonomies in which the number of classes range from hundreds of thousand as in Directory Mozilla to over a million in Wikipedia. As a result of ever increasing number of text documents and images from various sources, there is an immense need for automatic classification of documents in such large hierarchies. In this paper, we analyze the tradeoffs between the important characteristics of different classifiers employed in the top down fashion. The properties for relative comparison of these classifiers include, (i) accuracy on test instance, (ii) training time (iii) size of the model and (iv) test time required for prediction. Our analysis is motivated by the well known error bounds from learning theory, which is also further reinforced by the empirical observations on the publicly available data from the Large Scale Hierarchical Text Classfication Challenge. We show that by exploiting the data heterogenity across the large scale hierarchies, one can build an overall classification system which is approximately 4 times faster for prediction, 3 times faster to train, while sacrificing only 1% point in accuracy.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

Keywords

Hierarchical classification, Empirical Tradeoffs

1. INTRODUCTION

With an increasing amount of data from various sources such as web advertizing, social media and images, automatic classification of unseen data to one of tens of thousand target classes has caught the attention of the research community. In flat classification, no relationship is assumed between the target classes and $O(K)$ classifiers are learnt, one for each of the K classes. If some semantic structure exists among the classes, such as hierarchical, as in a rooted tree (Figure 1), a multi-class classifier is trained on each of the non-leaf node in the tree to distinguish between each of its children. For large scale classification, hierarchical strategies have two main advantages over flat classification: (i) to classify a test instance, one needs to evaluate only $O(\lg(K))$ classifiers, as against $O(K)$ for flat classification, and (ii) hierarchical classification may lead to better (in general comparable) predictive performance as compared to flat techniques [4].

In the context of large scale hierarchical classification, open challenges like the Pascal Large Scale Hierarchical Text Classification (LSHTC) [1] and Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [2] have been orgranized. In LSHTC for instance, the classes from the DMOZ and Wikipedia taxonomies are arranged in a rooted tree and directed acyclic graph respectively. The taxonomy thereby implicitly defines the semantic relationship among the classes. The publicly available DMOZ dataset contains around 400k training documents from the 27,875 target classes on the leaf nodes of the hierarchy tree with an extremely sparse representation involving 594,158 features. Outside of the LSHTC, various other approaches have also been proposed for large scale hierarchical classification, which have met with varying degrees of success (e.g., [1, 7]).

Previous approaches to large scale hierarchical categorization have mainly focused on the overall accuracy of the classifiers without taking into account other important factors such as: (i) training time to build the model, (ii) size of the model generated by fitting the parameters, and (iii) test time to predict the target class of a given test example.

We study here the tradeoffs between using generative models such as multinomial Naive Bayes, on one hand, and discriminative models such as Support Vector Machines (SVM) or Logistic Regression, on the other. In particular, we discuss the variation of training sample size from the root of hierarchy towards the leaves, which further determines the choice of model one might want to fit.

Another contribution of this work is to highlight a useful scenario in which one could combine both types of models in the larger hierarchy to get the best of both worlds. Large scale category hierarchies which occur in most practical and commercial applications, such as DMOZ used in our experi-

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

[2] http://www.image-net.org/challenges/LSVRC/2011/

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

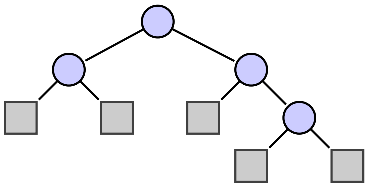

Figure 1: Example of a simple tree hierarchy, leaves are represented by squares

ments, are non-uniform across their entire structure. Therefore, to build an overall classification scheme, it is imperative to use classifiers which suit that particular local regime of operation. Empirical observations further demonstrate the interplay between various metrics of interest as we go from a fully discriminative setting to a fully generative framework.

We would also like to point out that the scope of this work is orthogonal to the large scale learning analysis by applying stochastic gradient descent [2] which essentially deals with binary classification in the context of large number of training examples. They stress on the fact that, in order to attain better training performance, one need not *fully* solve the optimization problem in learning the parameters and can stop the optimization process long before its convergence.

2. TRADEOFFS IN LARGE SCALE HIERARCHICAL CLASSIFICATION

In single-label multi-class hierarchical classification, the training set can be represented by $S = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$. In the context of text classification, $\mathbf{x}^{(i)} \in \mathcal{X}$ denotes the vector representation of document i in the input space $\mathcal{X} \subseteq \mathbb{R}^d$. Assuming that there are K classes denoted by the set $\mathcal{Y} = \{1 \ldots K\}$, the label $y^{(i)} \in \mathcal{Y}$ represents the class associated with the instance $\mathbf{x}^{(i)}$. The hierarchy in the form of rooted tree is given by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \supseteq \mathcal{Y}$ denotes the set of nodes of $\mathcal{G}$, and $\mathcal{E}$ denotes the set of edges with parent-to-child orientation. The leaves of the tree which usually forms the set of target classes is given by $\mathcal{Y} = \{u \in \mathcal{V} : \nexists v \in \mathcal{V}, (u, v) \in \mathcal{E}\}$.

In the above setup, given a new test instance $\mathbf{x}$, the goal is to predict the class $\widehat{y}$. This is typically done by making a sequence of predictions iteratively in a top-down fashion starting from the root until a leaf node is reached. At each non-leaf node $v \in \mathcal{V}$, a score $f_c(\mathbf{x}) \in \mathbb{R}$ is computed for each child c and the child $\widehat{c}$ with the maximum score is predicted i.e. $\widehat{c} = \underset{c:(v,c)\in\mathcal{E}}{\text{argmax}} f_c(\mathbf{x})$.

For our analysis, we focus on SVM and Multinomial Naive Bayes (NB) representing discriminative and generative models respectively. In SVM, $f_c(\mathbf{x})$ is modeled as a linear classifier such that $f_c(\mathbf{x}) = \mathbf{w}_c^t \mathbf{x}$. To learn an SVM-based discriminative classifier for node v, we solve the following optimization problem for each child c of v

$$\min_{\mathbf{w}_c, \boldsymbol{\xi}} \frac{\lambda}{2} ||\mathbf{w}_c||^2 + \sum_{i=1}^{n_v} \xi_i^2$$

The indices i above are such that $\forall i, 1 \leq i \leq n_v, y_i \in L_v$, were L_v denotes the set of leaves in the subtree rooted at node v and n_v denotes the number of training examples for which the root-to-leaf path passes through the node v. Furthermore, if $y_i \in L_c$ and $(v, c) \in \mathcal{E}$, then the constraints for the above optimization problem are given by, $\forall i$

$$\mathbf{w}_c^t \mathbf{x}_i \geq \mathbf{w}_{c'}^t \mathbf{x}_i - \xi_i, \ \forall c' \neq c \text{ s.t. } (v, c) \in \mathcal{E}, (v, c') \in \mathcal{E} \text{ and } \xi_i \geq 0$$

We use standard multinomial NB model in which predicted class is the one with maximum posterior probability, i.e.

$$\widehat{c} = \underset{c:(v,c)\in\mathcal{E}}{\text{argmax}} \Pr(c|\mathbf{x}), \text{ s.t. } \Pr(c|\mathbf{x}) \propto \Pr(c)\Pr(\mathbf{x}|c)$$

and the probabilities are replaced by their maximum likelihood estimates, taking Laplace smoothing into account.

2.1 Exploiting Data Heterogenity

For a multi-class classification problem at node v of the hierarchy, let d_v denote the dimensionality of the feature space and n_v denote the number of training documents for which the root-to-leaf path goes through node v. Let their ratio for node v be denoted by r_v, i.e. $r_v = \frac{d_v}{n_v}$.

In the context of large scale hierarchical classification, such as DMOZ, there is a wide spectrum over which r_v varies. For the classification problem corresponding to a node v at the top levels of the hierarchy tree, the ratio r_v is much higher as compared to its value for nodes at lower levels. Figure 2 shows the variation of average value of r_v for DMOZ dataset when plotted against the hierarchy levels. Each piece-wise linear curve in the plot corresponds to the class size range of the multi-class problem. Two important properties of the dataset, one of which follows from Figure 2, are: (i) The ratio r_v increases towards the leaves, and (ii) Almost 97% of the multi-class problems involve 2-15 classes.

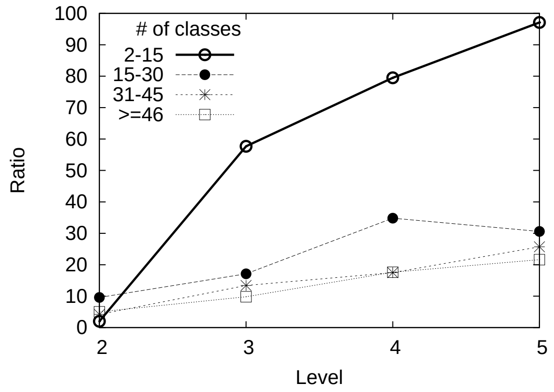

Figure 2: Variation in ratio of feature set size to training sample size with the hierarchy level. Level 2 corresponds to the children of root node and level 5 to the level that leads to leaves.

This shows that the nature of the learning problem posed is *different* in different parts of the hierarchy tree. We now present relevant results from statistical learning theory which are perfectly suited to address these problems [6]. Let f_G and f_D represent the classifiers learnt by fitting generative and discriminative model respectively and $f_{G,\infty}$ and $f_{D,\infty}$ be their corresponding asymptotic versions i.e. functions learnt when the sample size approaches infinity. Let $\varepsilon(.)$ be the function representing the generalization error of its argument. For a binary classification problem in d-dimensional feature space with n training examples, these results can essentially be summarized as follows:

1. $\varepsilon(f_{D,\infty}) \leq \varepsilon(f_{G,\infty})$;
2. $\varepsilon(f_G) \leq \varepsilon(f_{G,\infty}) + \delta_0$ if $n = \Omega(\ln(d))$;
3. $\varepsilon(f_D) \leq \varepsilon(f_{D,\infty}) + \delta_0'$ if $n = \Omega(d)$, for any fixed $\delta, \delta_0' > 0$ and $\Omega(.)$ denotes the big Omega notation

Argument 1 implies that in the regime of aymptotic operation, discriminative models should be preferred over generative models. Argument 2 and 3 suggest that generative classifiers approach their asymptotic performance with much lesser training data as compared to discriminative classifiers.

As a consequence of the above arguments, this implies the following design choices to build component classifiers for large scale hierarchical classification. We also briefly mention our observation for each of them in case of DMOZ data:

- On the nodes which are close to the root (including the root itself), we are close to the regime of asymptotic operation. Therefore using argument (1) from above, one should deploy discriminative classifiers such as SVM or logistic regression.
Observation for DMOZ : As shown in Figure 3, for level 1 and 2, SVM does indeed performs better and achieves much higher accuracy than NB classifier.

- Argument (2) above suggests that one should deploy NB classifier for the subproblems lower down the hierarchy since for *most* of the nodes, n is upper bounded by $\lg(d)$ i.e. $n = O(\lg(d))$.
Observation for DMOZ : As shown in Figure 3, for levels 4 and 5, NB cannot surpass the accuracy of SVM in this regime, which could be the result of argument (1). Importantly, however, the accuracy gap between the two classifiers is much smaller in this regime.

This indicates that, for lower levels in large hierarchy, NB is competetive to SVM and one can still employ NB instead of SVM, provided it can excel on metrics other than accuracy.

2.2 Adaptive Classifier Selection

From above observations for the DMOZ dataset, if prediction accuracy is the only criterion, then employing SVM over the entire hierarchy seems to be the classifier of choice. However, this comes with a few cons as well, which include: (i) more training time to train the classifiers, (ii) large size of the models built from the training data, (iii) due to which the models need to be read from hard disk every time for hierarchical predictions which leads to significant slowdown for prediction time. The NB classifier, on the other hand, does not suffer from these disadvantages. Moreover, due to compact models in this case, one can load all the classifiers of the hierarchy in the physical memory and can get massive speedup for prediction.

This leads us to the conclusion that, depending on the relative priority to satisfy the conflicting constraints of accuracy and run-time, we can get best of both models by combining SVM and NB classifiers in an adaptive way. For node v in the hierarchy, this can be achieved by using a threshold τ_v for the feature set size to sample size ratio r_v. The threshold value τ_v determines the choice of the classifier in the following way

$$\text{Classifier at node } v = \begin{cases} \text{Naive Bayes} & \text{if } r_v \geq \tau_v \\ \text{SVM} & \text{otherwise} \end{cases}$$

Property Name	Value
Total number of training examples	394,756
Size of the Overall Feature Space	594,158
Number of Target Classes ($\lvert\mathcal{Y}\rvert$)	27,875
Number of Nodes in the Hierarchy ($\lvert\mathcal{V}\rvert$)	35,449
Size of training file on Disk	586.3 MB
Depth of Hierarchy Tree	6
Total number of multiclass classifiers	7,574
Number of classifiers at depth 5	5,055

Table 1: Training Data Properties

The parameter $\tau = \{\tau_v\}, \forall v \in \mathcal{V}$, thus controls the tradeoff between accuracy of the overall classification system and the response time for training and prediction. Even though the above thresholding strategy is a simplification of the classifier selection criterion in section 2.1, it works well in practice as shown in our experiments and presented in more detail in section 4.

3. EXPERIMENTAL SETUP

The experiments were performed on a Linux system with 24GB physical memory and 1TB hard-disk. We use the publicly available DMOZ data set from the LSHTC, 2011. The dataset, after having been preprocessed by stemming and stopword removal, appears in the LibSVM format. Table 1 presents the numeric values corresponding to the important properties of the dataset. Since the average number of labels per document is 1.02, we consider it as single-label classification problem for our purpose.

We use Liblinear [3] to train the models for L2-regularized L2-loss support vector classification. The optimization problem was solved in the primal, since the dual formulation failed to converge for training classifier at the root node. The models are trained for all 7,574 non-leaf nodes in the hierarchy for One-Vs-All classification. For NB classifier, we implement the standard multinomial Naive Bayes using Laplace smoothing. Predictions are done in a top-down manner starting at the root node till the class corresponding to a leaf node is finally predicted.

Table 2 shows the different classification mechanisms to build the overall classifier, which include, (i) SVM classifier for the entire hierarchy, (ii) Adaptive classifier selection strategy based on threshold value, (iii) Static classifier selection by deploying NB classifier at lower levels, and finally (iv) NB classifier for the entire hierarchy. By employing SVM-only classification system, the accuracy (35.6%) is comparable to the best participant (38.8%) in LSHTC for the DMOZ track. However, we would like to point out that the objective of our work does not coincide with the participants' in the LSHTC challenge since the major focus of the challenge is on accuracy related metrics. As a result, some of the participants do not necessarily utilize the hierarchy completely as in [5] or may employ some postprocessing for higher accuracy. On the other hand, we take a more principled approach leading to a more robust and interpretable analysis which is also applicable to other large scale hierarchical classification problems involving more complex topologies such as directed acyclic graphs. Moreover, we aim to study the tradeoffs involving various constraints which

Model employed	Accuracy in %	Training Time (hours)	Test Time (secs)
SVM for entire hierarchy	35.6	35	20
Adaptive Selection, $\tau = 60$	35.2	22	12
Adaptive Selection, $\tau = 30$	34.7	12	5
SVM with NB for last level	32.4	14	4
NB for entire hierarchy	22.2	0.25	0.5

Table 2: Tradeoff between Prediction Accuracy in %, Total Training for entire dataset in hours, and Average Test Time per Instance in seconds

could be used to *tune* the desired behavior for a large scale hierarchical classification system.

4. RESULTS AND ANALYSIS

Table 2 shows the tradeoffs as we go from a fully discriminative framework to a fully generative one. When replacing the SVM classifiers (row 1) at the outer-most periphery of the hierarchy by NB (row 4), there is a 10% decrease in accuracy while the gain in prediction speed is close to 500%. This property could be leveraged to make robust real-time predictions such as for large scale Question-Answering systems or data stream environments which need real-time response for acceptable behavior. Also, there is an almost 3-fold improvement in training time as a result of this adaptation.

The gain in speed-up for training and test time is achieved as a result of more compact models built by NB as compared to SVM from same training data. All the NB models can, therefore, be loaded in the physical memory for predictions. For SVM, the total size of all the models is almost twice the physical memory size and hence the models for only the top two levels can be loaded in the physical memory.

The adaptive classifier selection as shown in row 2 and 3 of Table 2 was computed based on a uniform threshold value of $\tau_v = 60$ and $\tau_v = 30$, $\forall v \in \mathcal{V}$. Increasing the threshold value would select more SVM classifiers and thereby leading to better accuracy but slower training and test time. Decreasing it would correspond to more NB classifiers in the hierarchical framework, which leads to better run-time performance but lower accuracy.

Comparison between the adaptive classifier selection strategy and the static rule of applying NB classifier for the last level, rows 3 and 4 of Table 2, reveals another interesting observation. The prediction accuracy is noticeably higher by employing the adaptive strategy, for comparable values of training and prediction time.

Figure 3 shows the variation of difference in accuracy of SVM and NB classifiers when plotted against levels in the hierarchy. As per the arguments given in section 2.1, SVM outperforms NB at the levels near the root node of the hierarchy. However, NB catches up with SVM for the classifiers at level 4 and level 5 of the hierarchy but it is not able to surpass SVM accuracy. This could be due to argument (1), i.e. $\varepsilon(f_{D,\infty}) \leq \varepsilon(f_{G,\infty})$, which implies that asymptotic generalization performance of SVM is better than that of NB.

5. CONCLUSION AND FUTURE WORK

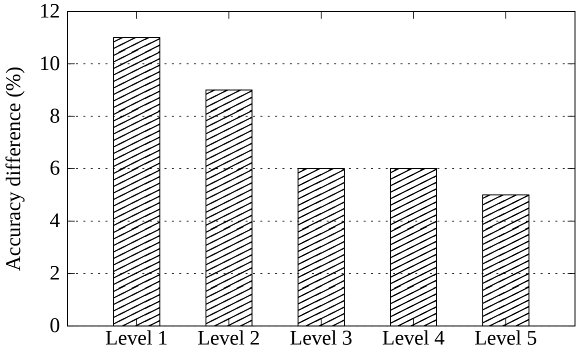

Figure 3: Difference of SVM and NB accuracy, (SVM - NB), in % for each hierarchy level. Level 1 corresponds to the root and level 5 to the level leading to leaves.

In this paper, we presented tradeoffs between conflicting constraints of prediction accuracy and computing resources which are crucial for the design of large scale hierarchical classification systems. Our analysis was based on utilizing the heterogeneity in large scale web directories, such as DMOZ, for designing effective local classifiers. We also presented an adaptive classifier selection strategy which can be employed to tune the extent of tradeoff. There are numerous avenues of further investigation, such as, (i) exploring more complex hierarchies such as graphs with cycles, (ii) addressing the data imbalance problems among classes more effectively, and (iii) extension to multi-label predictions.

6. ACKNOWLEDGEMENTS

This work was supported by the French National Research Agency(ANR). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ANR.

7. REFERENCES

[1] P. N. Bennett and N. Nguyen. Refined experts: improving classification in large taxonomies. In *In Proc. 32nd Int'l ACM SIGIR Conf. on Research and Dev. in Info. Retr.*, SIGIR 2009, pages 11–18.

[2] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *NIPS*, pages 161–168. 2008.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[4] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, pages 36–43, 2005.

[5] O. Madani and J. Huang. Large-scale many-class prediction via flat techniques. *Workshop on Large-Scale Hierarchical Text Classification at ECIR*, 2010.

[6] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Neural Information Processing Systems*, pages 841–848, 2001.

[7] G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *In Proc. 31st Int'l ACM SIGIR Conf. on Research and Dev. in Info. Retr.*, SIGIR '08, pages 619–626. ACM.