Boosting Ensembles of Structured Prediction Rules *

Corinna Cortes
Google Research
76 Ninth Avenue
New York, NY 10011
corinna@google.com

Vitaly Kuznetsov
Courant Institute
251 Mercer Street
New York, NY 10012
vitaly@cims.nyu.edu

Mehryar Mohri
Courant Institute & Google Research
251 Mercer Street
New York, NY 10012
mohri@cims.nyu.edu

Abstract

We present a boosting algorithm for structured prediction with theoretical guarantees for learning accurate ensembles of several structured prediction rules for which no prior knowledge is assumed.

1 Introduction

We study the problem of learning accurate ensembles of structured prediction experts. Ensemble methods are widely used in machine learning and have been shown to be often very effective [Breiman1996, Freund and Schapire1997, Smyth and Wolpert1999, MacKay1991, Freund et al.2004]. However, ensemble methods and their theory have been developed primarily for binary classification or regression tasks. Their techniques do not readily apply to structured prediction problems. While it is straightforward to combine scalar outputs for a classification or regression problem, it is less clear how to combine structured predictions such as phonemic pronunciation hypotheses, speech recognition lattices, parse trees, or alternative machine translations.

Ensemble structured prediction problems often arise in standard NLP tasks, such as part-of-speech tagging, as well as other applications such as optical character recognition, machine translation and computer vision, with structures or substructures varying with each task. We seek to tackle all of these problems simultaneously and consider the general setting where the label or output associated to an input $\mathbf{x} \in \mathcal{X}$ is a structure $\mathbf{y} \in \mathcal{Y}$ that can be decomposed and represented by l substructures $y^1, \ldots, y^l$. For the part-of-speech tagging example, $\mathbf{x}$ is a specific sentence and $\mathbf{y}$ is a corresponding sequence of tags. A natural choice for the substructures y^k is then the individual words forming $\mathbf{y}$.

We will assume that the loss function considered admits an additive decomposition over the substructures, as is common in structured prediction. We also assume access to a set of structured prediction experts $h_1, \ldots, h_p$ that we treat as black boxes. Given an input $\mathbf{x} \in \mathcal{X}$, each expert predicts a structure $h_j(\mathbf{x}) = (h_j^1(\mathbf{x}), \ldots, h_j^l(\mathbf{x}))$. The hypotheses h_j may have been derived using other machine learning algorithms or they may be based on carefully hand-crafted rules. Given a labeled training sample $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)$, our objective is to use the predictions of these experts to form an accurate ensemble.

A number of ensemble methods for structured prediction has been previously proposed in machine learning and natural language processing literature [Nguyen and Guo2007, Kocev et al.2013, Wang et al.2007, Collins and Koo2005, Zeman and Žabokrtský2005, Sagae and Lavie2006, Zhang et al.2009, Mohri et al.2008, Petrov2010, Fiscus1997]. Most of the references just mentioned do not give a rigorous theoretical justification for the techniques proposed. See [Cortes et al.2014a, Cortes et al.2014b] for the detailed overview. We are not aware of any prior theoretical analysis for the ensemble structured prediction problem we consider. Here, we present a boosting algorithm for learning ensembles of structured prediction rules that both

*This paper is a modified version of [Cortes et al.2014a, Cortes et al.2014b] to which we refer the reader for the proofs of the theorems stated and a more detailed discussion of our algorithms.

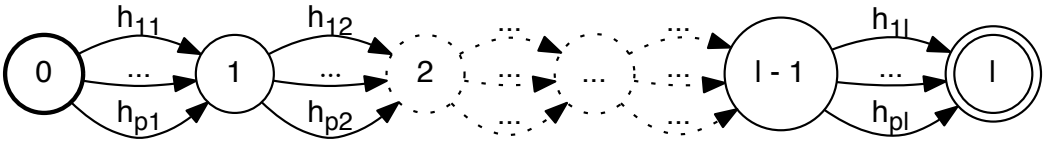

Figure 1: Finite automaton G of path experts.

perform well in practice and enjoy strong theoretical guarantees. We also refer the reader to [Cortes et al.2014a, Cortes et al.2014b] for another family of ensemble algorithms for structured prediction that also have good theoretical guarantees and performance.

We adopt a stadard supervised learning scenario, assuming that the learner receives a training sample $S = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)) \in \mathcal{X} \times \mathcal{Y}$ of m labeled points drawn i.i.d. according to the some distribution $\mathcal{D}$ used both for training and testing. We also assume that the learner has access to a set of p predictors $h_1, \ldots, h_p$ mapping $\mathcal{X}$ to $\mathcal{Y}$ to devise an accurate ensemble prediction. No other information is available to the learner about these p experts, in particular the way they have been trained or derived is not known to the learner. For a fixed $l \geq 1$, the quality of the predictions is measured by a loss function $L \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ that can be decomposed as a sum of loss functions $\ell_k \colon \mathcal{Y}_k \to \mathbb{R}_+$ over the substructure sets $\mathcal{Y}_k$, that is, for all $\mathbf{y} = (y^1, \ldots, y^l) \in \mathcal{Y}$ with $y^k \in \mathcal{Y}_k$ and $\mathbf{y}' = (y'^1, \ldots, y'^l) \in \mathcal{Y}$ with $y'^k \in \mathcal{Y}_k$,

$$L(\mathbf{y}, \mathbf{y}') = \sum_{k=1}^{l} \ell_k(y^k, y'^k). \tag{1}$$

We will assume in all that follows that the loss function L is bounded: $L(\mathbf{y}, \mathbf{y}') \leq M$ for all $(\mathbf{y}, \mathbf{y}')$ for some $M > 0$. A prototypical example of such loss functions is the normalized Hamming loss L_{Ham}, which is the fraction of substructures for which two labels $\mathbf{y}$ and $\mathbf{y}'$ disagree, thus in that case $\ell_k(y^k, y'^k) = \frac{1}{l} I_{y^k \neq y'^k}$ and $M = 1$.

2 Algorithm

Observe that each expert h_j induces a set of substructure hypotheses $h_j^1, \ldots, h_j^l$. One particular expert may be better at predicting the kth substructure while some other expert may be more accurate at predicting another substructure. Therefore, it is desirable to combine the substructure predictions of all experts to derive a more accurate prediction. This leads us to considering an acyclic finite automaton G such as that of Figure 1 which admits all possible sequences of substructure hypotheses, or, more generally, any acyclic finite automaton. An automaton such as G compactly represents a set of *path experts*: each path from the initial vertex 0 to the final vertex l is labeled with a sequence of substructure hypotheses $h_{j_1}^1, \ldots, h_{j_l}^l$ and defines a hypothesis which associates to input $\mathbf{x}$ the output $h_{j_1}^1(\mathbf{x}) \cdots h_{j_l}^l(\mathbf{x})$. We will denote by H the set of all path experts. We also denote by h each path expert defined by $h_{j_1}^1, \ldots, h_{j_l}^l$, with $j_k \in \{1, \ldots, p\}$, and denote by h^k its kth substructure hypothesis $h_{j_k}^k$. Our ensemble structure prediction problem can then be formulated as that of selecting the best path expert (or collection of path experts) in G. Note that, in general, the path expert selected does not coincide with any of the original experts $h_1, \ldots, h_p$.

In this section, we devise a boosting-style algorithm for our ensemble structured prediction problem. The variants of AdaBoost for multi-class classification such as AdaBoost.MH or AdaBoost.MR [Freund and Schapire1997, Schapire and Singer1999, Schapire and Singer2000] cannot be readily applied in this context. First, the number of classes to consider here is quite large, as in all structured prediction problems, since it is exponential in the number of substructures l. For example, in the case of the pronunciation problem where the number of phonemes for English is in the order of 50, the number of classes is 50^l. But, the objective function for AdaBoost.MH or AdaBoost.MR as well as the main steps of the algorithms include a sum over all possible labels, whose computational cost in this context would be prohibitive. Second, the loss function we consider is the normalized Hamming loss over the substructures predictions, which does not match the multi-class losses for

Algorithm 1 ESPBoost Algorithm

Inputs: $S = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m))$; set of experts $\{h_1, \ldots, h_p\}$
for $i = 1$ **to** m **and** $k = 1$ **to** l **do**
  $\mathcal{D}_1(i,k) \leftarrow \frac{1}{ml}$
end for
for $t = 1$ **to** T **do**
  $\mathsf{h}_t \leftarrow \operatorname{argmin}_{\mathsf{h} \in \mathsf{H}} \mathbb{E}_{(i,k)\sim\mathcal{D}_t}[\mathbf{1}_{\mathsf{h}^k(\mathbf{x}_i) \neq y_i^k}]$
  $\epsilon_t \leftarrow \mathbb{E}_{(i,k)\sim\mathcal{D}_t}[\mathbf{1}_{\mathsf{h}_t^k(\mathbf{x}_i) \neq y_i^k}]$
  $\alpha_t \leftarrow \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t}$
  $Z_t \leftarrow 2\sqrt{\epsilon_t(1-\epsilon_t)}$
  for $i = 1$ **to** m **and** $k = 1$ **to** l **do**
    $\mathcal{D}_{t+1}(i,k) \leftarrow \frac{\exp(-\alpha_t \rho(\widetilde{\mathsf{h}}_t^k, \mathbf{x}_i, \mathbf{y}_i))\mathcal{D}_t(i,k)}{Z_t}$
  end for
end for
Return $\widetilde{\mathsf{h}} = \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}_t$

the variants of AdaBoost.[1] Finally, the natural base hypotheses for this problem admit a structure that can be exploited to devise a more efficient solution, which of course was not part of the original considerations for the design of these variants of AdaBoost.

The predictor $\mathcal{H}_{\text{Boost}}$ returned by our boosting algorithm is based on a scoring function $\widetilde{\mathsf{h}}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which, as for standard ensemble algorithms such as AdaBoost, is a convex combination of base scoring functions $\widetilde{\mathsf{h}}_t$: $\widetilde{\mathsf{h}} = \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}_t$, with $\alpha_t \geq 0$. The base scoring functions used in our algorithm have the form $\widetilde{\mathsf{h}}_t(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^l \widetilde{\mathsf{h}}_t^k(\mathbf{x}, \mathbf{y})$. In particular, these can be derived from the path experts in H by letting $\mathsf{h}_t^k(\mathbf{x}, \mathbf{y}) = \mathbf{1}_{\mathsf{h}_t^k(\mathbf{x}) = y^k}$. Thus, the score assigned to $\mathbf{y}$ by the base scoring function $\widetilde{\mathsf{h}}_t$ is the number of positions at which $\mathbf{y}$ matches the prediction of path expert h_t given input $\mathbf{x}$. $\mathcal{H}_{\text{Boost}}$ is defined as follows in terms of $\widetilde{\mathsf{h}}$ or h_ts: $\mathcal{H}_{\text{Boost}}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}} \widetilde{\mathsf{h}}(\mathbf{x}, \mathbf{y})$ We remark that the analysis and algorithm presented in this section are also applicable with a scoring function that is the product of the scores at each substructure k as opposed to a sum, that is, $\widetilde{\mathsf{h}}(\mathbf{x}, \mathbf{y}) = \prod_{k=1}^l \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}_t^k(\mathbf{x}, \mathbf{y})$. This can be used for example in the case where the experts are derived from probabilistic models.

For any $i \in [1, m]$ and $k \in [1, l]$, we define the *margin of* $\widetilde{\mathsf{h}}^k$ *for point* $(\mathbf{x}_i, \mathbf{y}_i)$ by $\rho(\widetilde{\mathsf{h}}^k, \mathbf{x}_i, \mathbf{y}_i) = \widetilde{\mathsf{h}}^k(\mathbf{x}_i, y_i^k) - \max_{y^k \neq y_i^k} \widetilde{\mathsf{h}}^k(\mathbf{x}_i, y^k)$. We first derive an upper bound on the empirical normalized Hamming loss of a hypothesis $\mathcal{H}_{\text{Boost}}$, with $\widetilde{\mathsf{h}} = \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}_t$.

Lemma 1. *The following upper bound holds for the empirical normalized Hamming loss of the hypothesis* $\mathcal{H}_{\text{Boost}}$*:*

$$\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim S}[L_{\text{Ham}}(\mathcal{H}_{\text{Boost}}(\mathbf{x}), \mathbf{y})] \leq \frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \exp\Big(-\sum_{t=1}^T \alpha_t \rho(\widetilde{\mathsf{h}}_t^k, \mathbf{x}_i, \mathbf{y}_i)\Big).$$

The proof of this lemma as well as that of several other theorems related to this algorithm can be found in [Cortes et al.2014a]. In view of this upper bound, we consider the objective function $F\colon \mathbb{R}^N \to \mathbb{R}$ defined for all $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in \mathbb{R}^N$ by the right hand side in the bound of Lemma 1. F is a convex and differentiable function of $\boldsymbol{\alpha}$. Our algorithm, ESPBoost (Ensemble Structured Prediction Boosting), is defined by the application of coordinate descent to the objective F. Algorithm 1 shows the pseudocode of the ESPBoost.

Our *weak learning assumption* in this context is that there exists $\gamma > 0$ such that at each round, ϵ_t verifies $\epsilon_t < \frac{1}{2} - \gamma$. Note that, at each round, the path expert h_t with the smallest error ϵ_t can be determined easily and efficiently by first finding for each substructure k, the h_t^k that is the best with respect to the distribution weights $\mathcal{D}_t(i,k)$.

[1] [Schapire and Singer1999] also present an algorithm using the Hamming loss for multi-class classification, but that is a Hamming loss over the set of classes and differs from the loss function relevant to our problem. Additionally, the main steps of that algorithm are also based on a sum over all classes.

Table 1: Average Normalized Hamming Loss, ADS1 and ADS2.

	ADS1, $m = 200$	ADS2, $m = 200$
$\mathcal{H}_{\text{MVote}}$	$\mathbf{0.0197 \pm 0.00002}$	$\mathbf{0.2172 \pm 0.00983}$
$\mathcal{H}_{\text{ESPBoost}}$	$\mathbf{0.0197 \pm 0.00002}$	$\mathbf{0.2267 \pm 0.00834}$
$\mathcal{H}_{\text{SLE}}$	0.5641 ± 0.00044	0.2500 ± 0.05003
$\mathcal{H}_{\text{Rand}}$	0.1112 ± 0.00540	0.4000 ± 0.00018
Best h_j	0.5635 ± 0.00004	0.4000

We have derived both a margin-based generalization bound in support of the ESPBoost algorithm and a bound on the empirical margin loss. For any $\rho > 0$, define the empirical margin loss of $\mathcal{H}_{\text{Boost}}$ by the following: $\widehat{R}_\rho(\widetilde{\mathsf{h}}/\|\boldsymbol{\alpha}\|_1) = \frac{1}{ml}\sum_{i=1}^m \sum_{k=1}^l \mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k, \mathbf{x}_i, \mathbf{y}_i) \leq \rho \|\boldsymbol{\alpha}\|_1}$, where $\widetilde{\mathsf{h}}$ is the corresponding scoring function. The following theorem can be proven using the multi-class classification bounds of [Koltchinskii and Panchenko2002, Mohri et al.2012] as can be shown in [Cortes et al.2014a].

Theorem 2. *Let $\mathcal{F}$ denote the set of functions $\mathcal{H}_{\text{Boost}}$ with $\widetilde{\mathsf{h}} = \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}_t$ for some $\alpha_1, \ldots, \alpha_t \geq 0$ and $\mathsf{h}_t \in \mathsf{H}$ for all $t \in [1, T]$. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $\mathcal{H}_{\text{Boost}} \in \mathcal{F}$:*

$$\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[L_{\text{Ham}}(\mathcal{H}_{\text{Boost}}(\mathbf{x}), \mathbf{y})] \leq \widehat{R}_\rho\left(\frac{\widetilde{\mathsf{h}}}{\|\boldsymbol{\alpha}\|_1}\right) + \frac{2}{\rho l}\sum_{k=1}^l |\mathcal{Y}_k|^2 \mathfrak{R}_m(H^k) + \sqrt{\frac{\log \frac{l}{\delta}}{2m}},$$

where $\mathfrak{R}_m(H^k)$ denotes the Rademacher complexity of $H^k = \{\mathbf{x} \mapsto \widetilde{\mathsf{h}}_t^k : j \in [1, p], y \in \mathcal{Y}_k\}$.

This theorem provides a margin-based guarantee for convex ensembles such as those returned by ESPBoost. The following theorem further provides an upper bound on the empirical margin loss.

Theorem 3. *Let $\widetilde{\mathsf{h}}$ denote the scoring function returned by ESPBoost after $T \geq 1$ rounds. Then, for any $\rho > 0$, the following inequality holds: $\widehat{R}_\rho(\widetilde{\mathsf{h}}/\|\boldsymbol{\alpha}\|_1) \leq 2^T \prod_{t=1}^T \sqrt{\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}$.*

As in the case of AdaBoost [Schapire et al.1997], it can be shown that for $\rho < \gamma$, $\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho} \leq (1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} < 1$ and the right-hand side of this bound decreases exponentially with T.

3 Experiments

Here we present the results of experiments on some artificial data sets. For more extensive results, as well as, details of the experimental setup we refer the reader to [Cortes et al.2014a, Cortes et al.2014b]. We compared $\mathcal{H}_{\text{ESPBoost}}$ to on-line based approaches $\mathcal{H}_{\text{Rand}}$ and $\mathcal{H}_{\text{MVote}}$ from [Cortes et al.2014a, Cortes et al.2014b], as well as $\mathcal{H}_{\text{SLE}}$ algorithm of [Nguyen and Guo2007]. The results are summarised in Table 1.

4 Conclusion

We presented a broad analysis of the problem of ensemble structured prediction, including a boosting algorithm with learning guarantees and extensive experiments. Our results show that our algorithms, can result in significant benefits in several tasks, which can be of a critical practical importance. The boosting-style algorithm we presented can be enhanced using recent theoretical and algorithmic results on *deep boosting* [Cortes et al.2014c]. We also refer the reader to [Cortes et al.2014a, Cortes et al.2014b] for an exhaustive analysis of another family of on-line based algorithm for learning ensembles of structured prediction rules. These algorithms also enjoy good theoretical guarantees and perform well in practice.

Acknowledgments

We warmly thank our colleagues Francoise Beaufays and Fuchun Peng for kindly extracting and making available to us the pronunciation data sets, and Cyril Allauzen, Richard Sproat and Brian Roark for help with other data sets. This work was partly funded by the NSF award IIS-1117591 and the NSERC PGS D3 award.

References

[Breiman1996] Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

[Collins and Koo2005] Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

[Cortes et al.2014a] Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. 2014a. Ensemble methods for structured prediction. In *Proceedings of ICML*.

[Cortes et al.2014b] Corinna Cortes, Vitaly Kuznetsov, and Mehryar Mohri. 2014b. Learning ensembles of structured prediction rules. In *Proceedings of ACL*.

[Cortes et al.2014c] Corinna Cortes, Mehryar Mohri, and Umar Syed. 2014c. Deep boosting. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 2014)*.

[Fiscus1997] Jonathan G Fiscus. 1997. Post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *Proceedings of the 1997 IEEE ASRU Workshop*, pages 347–354, Santa Barbara, CA.

[Freund and Schapire1997] Y. Freund and R. Schapire. 1997. A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[Freund et al.2004] Yoav Freund, Yishay Mansour, and Robert E. Schapire. 2004. Generalization bounds for averaged classifiers. *The Annals of Statistics*, 32:1698–1722.

[Kocev et al.2013] D. Kocev, C. Vens, J. Struyf, and S. Deroski. 2013. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, March.

[Koltchinskii and Panchenko2002] Vladmir Koltchinskii and Dmitry Panchenko. 2002. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30.

[MacKay1991] David J. C. MacKay. 1991. *Bayesian methods for adaptive models*. Ph.D. thesis, California Institute of Technology.

[Mohri et al.2008] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2008. Speech recognition with weighted finite-state transducers. In *Handbook on Speech Processing and Speech Communication, Part E: Speech recognition*. Springer-Verlag.

[Mohri et al.2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. The MIT Press.

[Nguyen and Guo2007] N. Nguyen and Y. Guo. 2007. Comparison of sequence labeling algorithms and extensions. In *Proceedings of ICML*, pages 681–688.

[Petrov2010] Slav Petrov. 2010. Products of random latent variable grammars. In *HLT-NAACL*, pages 19–27.

[Sagae and Lavie2006] K. Sagae and A. Lavie. 2006. Parser combination by reparsing. In *Proceedings of HLT/NAACL*, pages 129–132.

[Schapire and Singer1999] Robert E. Schapire and Yoram Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336.

[Schapire and Singer2000] Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168.

[Schapire et al.1997] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. 1997. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML*, pages 322–330.

[Smyth and Wolpert1999] Padhraic Smyth and David Wolpert. 1999. Linearly combining density estimators via stacking. *Machine Learning*, 36:59–83, July.

[Wang et al.2007] Q. Wang, D. Lin, and D. Schuurmans. 2007. Simple training of dependency parsers via structured boosting. In *Proceedings of IJCAI 20*, pages 1756–1762.

[Zeman and Žabokrtský2005] D. Zeman and Z. Žabokrtský. 2005. Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of IWPT 9*, pages 171–178.

[Zhang et al.2009] H. Zhang, M. Zhang, C. Tan, and H. Li. 2009. K-best combination of syntactic parsers. In *Proceedings of EMNLP: Volume 3*, pages 1552–1560.