Structural Online Learning

Mehryar Mohri[1,2] and Scott Yang[2]

[1] Google Research, 111 8th Avenue, New York, NY 10011,
mohri@cs.nyu.edu,
[2] Courant Institute, 251 Mercer Street, New York, NY, 10012,
yangs@cims.nyu.edu

Abstract. We study the problem of learning ensembles in the online setting, when the hypotheses are selected out of a base family that may be a union of possibly very complex sub-families. We prove new theoretical guarantees for the online learning of such ensembles in terms of the sequential Rademacher complexities of these sub-families. We also describe an algorithm that benefits from such guarantees. We further extend our framework by proving new structural estimation error guarantees for ensembles in the batch setting through a new data-dependent online-to-batch conversion technique, thereby also devising an effective algorithm for the batch setting which does not require the estimation of the Rademacher complexities of base sub-families.

1 Introduction

Ensemble methods are powerful techniques in machine learning for combining several predictors to define a more accurate one. They include notable methods such as bagging and boosting [4, 11], and they have been successfully applied to a variety of scenarios including classification and regression.

Standard ensemble methods such as AdaBoost and Random Forests select base predictors from some hypothesis set $\mathcal{H}$, which may be the family of boosting stumps or that of decision trees with some limited depth. More complex base hypothesis sets may be needed to tackle some difficult modern tasks. At the same time, learning bounds for standard ensemble methods suggest a risk of overfitting when using very rich hypothesis sets, which has been further observed empirically [10, 17].

Recent work in the batch setting has shown, however, that learning with such complex base hypothesis sets is possible using the *structure* of $\mathcal{H}$, that is its decomposition into subsets $\mathcal{H}_k$, $k = 1, \ldots, p$, of varying complexity. In particular, in [8], we introduced a new ensemble algorithm, *DeepBoost*, which we proved benefits from finer learning guarantees when using rich families as base classifier sets. In DeepBoost, the decisions in each iteration of which classifier to add to the ensemble and which weight to assign to that classifier depend on the complexity of the sub-family $\mathcal{H}_k$ to which the classifier belongs. This can be viewed as integrating the principle of structural risk minimization to each iteration of boosting.

This paper extends the *structural learning* idea of incorporating model selection in ensemble methods to the online learning setting. Specifically, we address the question: can one design ensemble algorithms for the online setting that admit strong guarantees even when using a complex $\mathcal{H}$? In Section 3, we first present a theoretical result guaranteeing the existence of a randomized algorithm that can compete against the best ensemble in $\mathcal{H}$ efficiently when this ensemble does not rely *too heavily* on complex base hypotheses. Motivated by this theory, we then design an online algorithm that benefits from such guarantees, for a wide family of hypotheses sets (Section 4). Finally, in Section 5, we further extend our framework by proving new structural estimation error guarantees for ensembles in the batch setting through a new data-dependent online-to-batch conversion technique. This also provides an effective algorithm for the batch setting which does not require the estimation of the Rademacher complexities of base hypothesis sets $\mathcal{H}_k$.

2 Notation and preliminaries

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space. Let $L_t \colon \mathcal{Y} \to \mathbb{R}_+$ be a loss function. The online learning framework that we study is a sequential prediction setting that can be described as follows. At each time $t \in [1, T]$, the learner (or algorithm $\mathcal{A}$) receives an input instance x_t which he uses to select a hypothesis $h_t \in \mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and make a prediction $h_t(x_t)$. The learner then incurs the loss $L_t(h_t(x_t))$ based on the loss function L_t chosen by an adversary. The objective of the learner is to minimize his regret over T rounds, that is the difference of his cumulative loss $\sum_{t=1}^T L_t(h_t(x_t))$ and that of the best function in some benchmark hypothesis set $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$:

$$\mathrm{Reg}_T(\mathcal{A}) = \sum_{t=1}^T L_t(h_t(x_t)) - \min_{h \in \mathcal{F}} \sum_{t=1}^T L_t(h(x_t)).$$

In what follows, $\mathcal{F}$ will be assumed to coincide with $\mathcal{H}$, unless explicitly stated otherwise. The learner's algorithm may be randomized, in which case, at each round t, the learner draws hypothesis h_t from the distribution π_t he has defined at that round. The regret is then the difference between the expected cumulative loss and the expected cumulative loss of the best-in-class hypothesis: $\mathrm{Reg}_T(\mathcal{A}) = \sum_{t=1}^T \mathbb{E}[L_t(h_t(x_t))] - \min_{h \in \mathcal{H}} \sum_{t=1}^T \mathbb{E}[L_t(h(x_t))]$.

Clearly, the difficulty of the learner's regret minimization task depends on the richness of the competitor class $\mathcal{H}$. The more complex $\mathcal{H}$ is, the smaller the loss of the best function in $\mathcal{H}$ and thus the harder the learner's benchmark. This complexity can be captured by the notion of *sequential Rademacher complexity* introduced by [16]. Let $\mathcal{H}$ be a set of functions from $\mathcal{X}$ to $\mathbb{R}$. The sequential Rademacher complexity of a hypothesis $\mathcal{H}$ is denoted by $\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H})$ and defined by

$$\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}) = \frac{1}{T} \sup_{\mathbf{x}} \mathbb{E}\left[\sup_{h \in \mathcal{H}} \sum_{t=1}^T \sigma_t h(x_t(\boldsymbol{\sigma})) \right], \tag{1}$$

where the supremum is taken over all $\mathcal{X}$-*valued complete binary trees* of depth T and where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_T)$ is a sequence of i.i.d. Rademacher variables, each taking values in $\{\pm 1\}$ with probability $\frac{1}{2}$. Here, an $\mathcal{X}$-valued complete binary tree $\mathbf{x}$ is defined as a sequence $(x_1, \ldots, x_T)$ of mappings where $x_t \colon \{\pm 1\}^{t-1} \to \mathcal{X}$. The root x_1 can be thought of as some constant in $\mathcal{X}$. The left child of the root is $x_2(-1)$ and the right child is $x_2(1)$. A path in the tree is $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{T-1})$. To simplify the notation, we write $x_t(\boldsymbol{\sigma})$ instead of $x_t(\sigma_1, \ldots, \sigma_{t-1})$. The sequential Rademacher complexity can be interpreted as the online counterpart of the standard Rademacher complexity widely used in the analysis of batch learning [1, 13]. It has been used by [16] and [15] both to derive attainability results for some regret bounds and to guide the design of new online algorithms.

3 Theoretical guarantees for structural online learning

In this section, we present learning guarantees for structural online learning in binary classification. Hence, for any $t \in [1, T]$, the loss incurred at each time t by hypothesis h is $L_t(h(x_t)) = 1_{\{y_t h(x_t) < 0\}}$, with $y_t \in \mathcal{Y} = \{\pm 1\}$.

A *randomized player strategy* $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_T)$ for a sequence of length T is a sequence of mappings $\pi_t \colon (\mathcal{X} \times \mathcal{Y})^{t-1} \to \mathcal{P}_{\mathcal{H}}$, $t \in [T]$, where $\mathcal{P}_{\mathcal{H}}$ is the family of distributions over $\mathcal{H}$. Thus, $\pi_t((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$ is the distribution according to which the player selects a hypothesis $h \in \mathcal{H}$ at time t and which also depends on the past sequence $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$ played against the adversary.

The following shows the existence of a randomized strategy that benefits from a margin-based regret guarantee in the online setting. This can be viewed as the counterpart of the classical margin-based learning bounds in the batch setting given by [13].

Theorem 1 (Proposition 25 [16]). *For any function class $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ of functions bounded by one, there exists a randomized player strategy given by $\boldsymbol{\pi}$ such that for any sequence $z_1, \ldots, z_T$ played by the adversary, $z_t = (x_t, y_t) \in \mathcal{X} \times \{\pm 1\}$, the following inequality holds:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h_t \sim \pi_t}[1_{\{y_t h_t(x_t) < 0\}}]\right] \\ \leq \inf_{\gamma > 0}\left\{\inf_{h \in \mathcal{H}} \frac{1}{T}\sum_{t=1}^{T} 1_{\{y_t h(x_t) < \gamma\}} + \frac{4}{\gamma}\mathcal{R}_T^{\mathrm{seq}}(\mathcal{H}) + \frac{1}{\sqrt{T}}\left(3 + \log\log\frac{1}{\gamma}\right)\right\}.$$

We are not explicitly indicating the dependency of π_t on $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$ to alleviate the notation. The theorem gives a guarantee for the expected error of a randomized strategy in terms of the empirical margin loss and the sequential Rademacher complexity of $\mathcal{H}$ scaled by γ for the best choice of $h \in \mathcal{H}$ and the best confidence margin γ. As with standard margin bounds, this is subject to a trade-off: for a larger γ the empirical margin loss is larger, while a smaller value of γ increases the complexity term. The result gives a very favorable guarantee when there exists a relatively large γ for which the empirical margin loss of the best h is relatively small.

While this result is remarkable for characterizing learnability against the best-in-class hypothesis, it does not identify and take advantage of any structure in the hypothesis set. The structural margin bound that we prove next specifically provides a guarantee that exploits the scenario where the hypothesis set admits a decomposition $\mathcal{H} = \cup_{k=1}^{p} \mathcal{H}_k$. For any $q \in \mathbb{N}$, we will denote by $\mathit{\Delta}_q$ the probability simplex in $\mathbb{R}^q$ and by $\mathrm{conv}(\mathcal{H})$ the convex hull of $\mathcal{H}$.

Theorem 2. *Let $\mathcal{H} \subset [-1,1]^{\mathcal{X}}$ be a family of functions admitting a decomposition $\mathcal{H} = \bigcup_{k=1}^{p} \mathcal{H}_k$. Then, there exists a randomized player strategy given by π on $\mathrm{conv}(\mathcal{H})$ such that for any sequence $((x_t, y_t))_{t\in[T]}$ in $\mathcal{X} \times \{\pm 1\}$, the following inequality holds:*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{f_t\sim\pi_t}\left[1_{\{y_t f_t(x_t)<0\}}\right]\right] \le \inf_{\gamma>0}\left\{\inf_{\substack{f=\sum_{i=1}^{q}\alpha_i h_i\in\,\mathrm{conv}(\mathcal{H})\\ \boldsymbol{\alpha}\in\mathit{\Delta}_q,\, h_i\in\mathcal{H}_{k(h_i)}}} \frac{1}{T}\sum_{t=1}^{T} 1_{\{y_t f(x_t)<\gamma\}} + \frac{6}{\gamma}\sum_{i=1}^{q}\alpha_i \mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_{k(h_i)}) + \widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\sqrt{\frac{\log p}{T}}\right)\right\},$$

where $k(h_i) \in [1,p]$ is defined to be the smallest index k such that $h_i \in \mathcal{H}_k$ and $q \in \mathbb{N}$ so that f is an arbitrary element in the convex hull.

The theorem extends the margin bound of Theorem 1 given for a single hypothesis set to a guarantee for the convex hull of p hypothesis sets. Observe that, remarkably, the complexity term depends on the mixture weights α_i and hypotheses h_i defining the the best-in-class hypothesis $f = \sum_{i=1}^{q} \alpha_i h_i$. The complexity term is an $\boldsymbol{\alpha}$-average of the sequential Rademacher complexities. Thus, the theorem shows the existence of a randomized strategy π that achieves a favorable guarantee so long as the best-in-class hypothesis $f \in \mathrm{conv}(\mathcal{H})$ admits a decomposition for which the complexity term is relatively small, which directly depends on the amount of mixture weight assigned to more complex $\mathcal{H}_k$s versus less complex ones in the decomposition of f.

From a proof standpoint, it is enticing to use the fact that the sequential Rademacher complexity of a hypothesis set does not increase upon taking the convex hull. While this property yields an interesting result itself, it is not sufficiently fine for deriving the result of Theorem 2: in short, the resulting guarantee is then in terms of the maximum of the sequential Rademacher complexities instead of their $\boldsymbol{\alpha}$-average.

Proof. Fix $n \geq 1$. For any p-tuple of non-negative integers $\mathbf{N} = (N_1, \ldots, N_p) \in \mathbb{N}^p$ with $|\mathbf{N}| = \sum_{k=1}^{p} N_k = n$, consider the following family of functions:

$$G_{\mathcal{H},\mathbf{N}} = \left\{ \frac{1}{n}\sum_{k=1}^{p}\sum_{j=1}^{N_k} h_{k,j} \;\middle|\; \forall (k,j) \in [1,p] \times [N_k], h_{k,j} \in H_k \right\}.$$

By the sub-additivity of the supremum operator, the sequential Rademacher complexity of $\mathcal{H}$ can be upper bounded as follows:

$$\begin{aligned}\mathfrak{R}_T^{\mathrm{seq}}(G_{\mathcal{H},\mathbf{N}}) &= \frac{1}{T}\sup_{\mathbf{x}}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h\in G_{\mathcal{H},N}}\sum_{t=1}^{T}\frac{1}{n}\sum_{k=1}^{p}\sum_{j=1}^{N_k}h_{k,j}(x_t(\boldsymbol{\sigma}))\sigma_t\right]\\ &\leq \frac{1}{T}\frac{1}{n}\sum_{k=1}^{p}\sum_{j=1}^{N_k}\sup_{\mathbf{x}}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{h_{k,j}\in\mathcal{H}_k}\sum_{t=1}^{T}h_{k,j}(x_t(\boldsymbol{\sigma}))\sigma_t\right] = \frac{1}{n}\sum_{k=1}^{p}N_k\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_k).\end{aligned}$$

In view of this inequality and the margin bound of Proposition 1, for any $G_{\mathcal{H},\mathbf{N}}$, there exists a player strategy $\pi^{\mathbf{N}}$ such that

$$\begin{aligned}&\frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T}\mathop{\mathbb{E}}_{f_t\sim\pi_t^{\mathbf{N}}}[1_{\{y_tf_t(x_t)<0\}}]\right]\\ &\quad\leq \inf_{\gamma>0}\left\{\inf_{g\in G_{\mathcal{H},\mathbf{N}}}\frac{1}{T}\sum_{t=1}^{T}1_{\{y_tg(x_t)<\gamma\}}+\frac{4}{\gamma}\frac{1}{n}\sum_{k=1}^{p}N_k\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_k)+\frac{3+\log\log\frac{1}{\gamma}}{\sqrt{T}}\right\}.\end{aligned}$$

Now, let $\pi^{\exp}$ denote a randomized weighted majority strategy $\pi^{\exp}$ with the $\pi^{\mathbf{N}}$ strategies serving as experts [14]. Since there are at most p^n p-tuples $\mathbf{N}$ with $|\mathbf{N}|=n$, the regret of this randomized weighted majority strategy is bounded by $2\sqrt{T\log(p^n)}$ (see [6]). Thus, the following guarantee holds for the strategy $\pi^{\exp}$:

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathop{\mathbb{E}}_{f_t\sim\pi_t^{\exp}}[1_{\{y_tf_t(x_t)<0\}}]\right]\leq\inf_{|\mathbf{N}|=n}\mathbb{E}\left[\sum_{t=1}^{T}\mathop{\mathbb{E}}_{f_t\sim\pi_t^{\mathbf{N}}}[1_{\{y_tf_t(x_t)<0\}}]\right]+\sqrt{4Tn\log p}.$$

In view of that, we can write

$$\begin{aligned}&\frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T}\mathop{\mathbb{E}}_{f_t\sim\pi_t^{\exp}}[1_{\{y_tf_t(x_t)<0\}}]\right]\\ &\leq\inf_{\gamma>0}\left\{\inf_{\substack{g\in G_{\mathcal{H},\mathbf{N}}\\|\mathbf{N}|=n}}\frac{1}{T}\sum_{t=1}^{T}1_{\{y_tg(x_t)<\gamma\}}+\frac{4}{\gamma}\frac{1}{n}\sum_{k=1}^{p}N_k\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_k)\right\} \qquad (2)\\ &\quad+\frac{3+\log\log\frac{1}{\gamma}}{\sqrt{T}}+2\sqrt{\frac{n\log p}{T}}\\ &=\inf_{\gamma>0}\left\{\inf_{\substack{g=\frac{1}{n}\sum_{i=1}^{q}n_ih_i\\h_i\in\mathcal{H}_{k(h_i)}}}\frac{1}{T}\sum_{t=1}^{T}1_{\{y_tg(x_t)<\gamma\}}+\frac{4}{\gamma}\frac{1}{n}\sum_{i=1}^{q}n_i\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_{k(h_i)})\right\}\\ &\qquad+\frac{3+\log\log\frac{1}{\gamma}}{\sqrt{T}}+2\sqrt{\frac{n\log p}{T}}. \qquad (3)\end{aligned}$$

Now, fix $(h_1,\ldots,h_q)$. Any $\boldsymbol{\alpha}\in\Delta_q$ defines a distribution over $h_1,\ldots,h_q$. Sampling according to $\boldsymbol{\alpha}$ and averaging leads to functions g of the form $g=\frac{1}{n}\sum_{i=1}^{q}n_ih_i$

for some q-tuple $\mathbf{n} = (n_1, \ldots, n_q)$ with $|\mathbf{n}| = n$. Let $f = \sum_{i=1}^{q} \alpha_i h_i$ for some $\boldsymbol{\alpha} \in \Delta_q$. By the union bound, we can write, for any $\gamma > 0$ and (x_t, y_t),

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[1_{y_t g(x_t) < \gamma}\right] &= \Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t g(x_t) < \gamma\right] = \Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t g(x_t) - y_t f(x_t) + y_t f(x_t) < \gamma\right] \\
&\leq \Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t g(x_t) - y_t f(x_t) < -\tfrac{\gamma}{2}\right] + \Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t f(x_t) < \tfrac{3\gamma}{2}\right] \\
&= \Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t g(x_t) - y_t f(x_t) < -\tfrac{\gamma}{2}\right] + 1_{y_t f(x_t) < \frac{3\gamma}{2}}.
\end{aligned}$$

For any $\gamma > 0$ and (x_t, y_t), by Hoeffding's inequality, the following holds:

$$\Pr_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[y_t g(x_t) - y_t f(x_t) < -\tfrac{\gamma}{2}\right] \leq e^{\frac{-n\gamma^2}{8}}.$$

Plugging this inequality back into the previous one gives:

$$\mathop{\mathbb{E}}_{\mathbf{n} \sim \boldsymbol{\alpha}} \left[1_{y_t g(x_t) < \gamma}\right] \leq e^{\frac{-n\gamma^2}{8}} + 1_{y_t f(x_t) < \frac{3\gamma}{2}}. \tag{4}$$

Fix $\gamma > 0$ and $h_1 \in \mathcal{H}_{k(h_1)}, \ldots, h_q \in \mathcal{H}_{k(h_q)}$ in inequality 2. Then, taking the expectation over $\boldsymbol{\alpha}$ of both sides of the inequality and using inequality 4 combined with $\mathbb{E}[\frac{n_i}{n}] = \alpha_i$, we obtain that for any $\gamma > 0$, any $h_1 \in \mathcal{H}_{k(h_1)}, \ldots, h_q \in \mathcal{H}_{k(h_q)}$, and any $\boldsymbol{\alpha} \in \Delta_q$, the following holds for $f = \sum_{i=1}^{q} \alpha_i h_i$:

$$\begin{aligned}
\frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{f_t \sim \pi_t^{\exp}} [1_{\{y_t f_t(x_t) < 0\}}]\right] \leq \frac{1}{T} \sum_{t=1}^{T} 1_{y_t f(x_t) < \frac{3\gamma}{2}} + \frac{4}{\gamma} \sum_{i=1}^{q} \alpha_i \mathfrak{R}_T^{\text{seq}}(\mathcal{H}_{k(h_i)}) \\
+ e^{\frac{-n\gamma^2}{8}} + \frac{3 + \log\log\frac{1}{\gamma}}{\sqrt{T}} + 2\sqrt{\frac{n \log p}{T}}.
\end{aligned}$$

Thus, this inequality holds for any $\gamma > 0$, any $n \geq 1$, and any $f = \sum_{i=1}^{q} \alpha_i h_i \in \operatorname{conv}(\mathcal{H})$. Choosing $n = \left\lceil \frac{4}{\gamma^2} \log \frac{\gamma^2 T}{16 \log p} \right\rceil$ and replacing $\frac{3\gamma}{2}$ by γ yields

$$\begin{aligned}
&\frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{f_t \sim \pi_t^{\exp}} [1_{\{y_t f_t(x_t) < 0\}}]\right] \\
&\leq \inf_{\gamma > 0} \Bigg\{ \inf_{\substack{f = \sum_{i=1}^{q} \alpha_i h_i \in \operatorname{conv}(\mathcal{H}) \\ \boldsymbol{\alpha} \in \Delta_q, h_i \in \mathcal{H}_{k(h_i)}}} \frac{1}{T} \sum_{t=1}^{T} 1_{y_t f(x_t) < \gamma} + \frac{6}{\gamma} \sum_{i=1}^{q} \alpha_i \mathfrak{R}_T^{\text{seq}}(\mathcal{H}_{k(h_i)}) \\
&\quad + 6\sqrt{\frac{\log p}{\gamma^2 T}} + 6\sqrt{\left\lceil \frac{1}{\gamma^2} \log\left[\frac{\gamma^2 T}{36 \log p}\right] \right\rceil \frac{\log p}{T}} + \frac{3 + \log\log\frac{3}{2\gamma}}{\sqrt{T}} \Bigg\},
\end{aligned}$$

which completes the proof. □

4 Algorithms for structural online learning

While Theorem 2 proves the existence of a randomized strategy with favorable structural learning guarantees, it does not explicitly define one. In this section, we give a general algorithm that benefits from the structural online learning bound above.

In what follows, we will fix an arbitrary decomposition of the function class: $\mathcal{H} = \cup_{k=1}^{p} \mathcal{H}_k$. Moreover, we will assume that this decomposition $(\mathcal{H}_k)_{k=1}^{p}$ is *structurally online linear-learnable* in the sense that for any subset $\mathcal{H}_k \subset \mathcal{H}$, $k \in [1, p]$, there exists an online learning algorithm $\mathcal{A}_k$ such that for any time horizon T and every sequence $(x_t, L_t)_{t=1}^{T}$ where L_t is a linear loss function bounded by 1, $\mathcal{A}_k$ selects a sequence of functions $(h_t)_{t=1}^{T}$ satisfying $\sum_{t=1}^{T} L_t(h_t(x_t)) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} L_t(h(x_t)) = \mathrm{Reg}_{\mathcal{H}_k, T}(\mathcal{A}_k) = o(T)$. For instance, if $\mathcal{H}$ is finite, every decomposition is structurally online linear-learnable, which can be seen by applying a potential-based algorithm [6]. Note that the notion of structural online linear-learnability is a slight generalization of the concept introduced by [2].

We will also follow in this section the standard method of using a convex surrogate for the zero-one loss function. The will enable us to design algorithms that are both deterministic and can be used to achieve new structural PAC guarantees in the batch setting (the latter will be seen in Section 5). Let $\Phi \colon \mathbb{R} \to \mathbb{R}$ be any convex loss function upper bounding the zero-one loss. We further assume that Φ is G-Lipschitz. One standard example is the hinge loss, $\Phi(x) = (1-x)1_{\{x \leq 1\}}$, which is 1-Lipschitz. Our goal is to design algorithms $\mathcal{A}$ that guarantee structural upper bounds on the following regret term: $\mathrm{Reg}_T(\mathcal{A}) = \max_{h \in \mathrm{conv}(\mathcal{H})} \sum_{t=1}^{T} \Phi(y_t h_t(x_t)) - \Phi(y_t h(x_t))$. For any x, we will denote by $\Phi'(x)$ an arbitrary element of the subgradient of Φ at x.

4.1 SOL.Boost algorithm

At first glance, the structural learning bound of Theorem 2 seems unwieldy since the convex combination of the best-in-class hypothesis is not necessarily well-ordered with respect to the decomposition of the hypothesis set. Moreover, the proof of Theorem 2 is based on the existence of online learning algorithms for different subclasses of functions to which a meta-algorithm for learning with *experts* is applied. This is instructive, but it is also computationally infeasible because there are exponentially many experts in the proof.

Addressing the well-ordering issue and the computational problem will be essential to our algorithmic design. Towards the first point, we can "re-organize" the best-in-class hypothesis by writing:

$$\begin{aligned}\sum_{i=1}^{q} \alpha_i^* \mathfrak{R}_T(\mathcal{H}_{k(i^*)}) &= \sum_{i=1}^{q} \sum_{k=1}^{p} 1_{\{k(i^*)=k\}} \alpha_i^* \mathfrak{R}_T(\mathcal{H}_{k(i^*)}) \\ &= \sum_{k=1}^{p} \sum_{i=1}^{q} 1_{\{k(i^*)=k\}} \alpha_i^* \mathfrak{R}_T(\mathcal{H}_k) = \sum_{k=1}^{p} \gamma_k^* \mathfrak{R}_T(\mathcal{H}_k).\end{aligned}$$

This suggests that learning against the convex hull of a hypothesis class with a structural decomposition can be equivalently cast as learning against each of its individual substructures along with some new set of convex weights. We will use

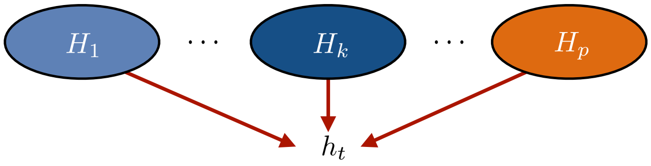

Fig. 1. Illustration of SOL.Boost Algorithm. The algorithm incorporates a meta-algorithm that measures the progress of each base algorithm. Note from the pseudo-code that the base algorithms are not assigned their true losses, but instead new hallucinated losses.

this observation by applying an (efficient) experts-type algorithm to learn these new weights instead of the original weights from the best convex combination.

However, learning against each of the individual substructures proves to be a challenge in and of itself, since typical online learning algorithms, such as the weighted majority algorithm [14], are able to provide guarantees against only the single "best-in-class" hypothesis. Direct application of the experts algorithm on top of typical online learning algorithms for each $\mathcal{H}_k$ will only produce a regret guarantee against comparators of the form $\sum_{k=1}^p \alpha_k^* h_k^*$, $h_k^* \in \mathcal{H}_k$, $\boldsymbol{\alpha}^* \in \Delta_p$. On the other hand, Theorem 2 guarantees the existence of an algorithm that can attain a structural regret bound against arbitrary convex combinations in $\mathcal{H}$, including those that contain multiple base hypotheses from a single substructure $\mathcal{H}_k$. To attain this type of guarantees, we will linearize the loss and *hallucinate* different losses for each of the base online linear learning algorithms so that they learn well against the convex hull of each subclass.

Our algorithm, SOL.Boost, incorporates these two ideas to produce a guarantee in the form of the one given in Theorem 2. Figure 1 presents an illustration of the algorithm.

Theorem 3. *Let $\mathcal{H} \subset [-1,1]^{\mathcal{X}}$ be a hypothesis set admitting the decomposition $\mathcal{H} = \cup_{k=1}^p \mathcal{H}_k$ that is structurally online linear-learnable. For each $k \in [1,p]$, let $\mathcal{A}_k$ be an online algorithm that can minimize the regret of linear loss functions against $\mathcal{H}_k$, with regret $\mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)$ over T rounds. Let $\Phi\colon \mathbb{R} \to \mathbb{R}$ be a G-Lipschitz convex upper bound on the zero-one loss. Then, SOL.Boost, initialized with $\eta < \sqrt{\frac{\log p}{T}}$, outputs a sequence of hypotheses $(h_t)_{t=1}^T$ that satisfies the following regret bound for any $(x_t, y_t)_{t=1}^T$:*

$$\sum_{t=1}^T \Phi(y_t h_t(x_t)) \leq \inf_{\substack{f=\sum_{i=1}^q \alpha_i h_i \in \mathrm{conv}(\mathcal{H}) \\ \boldsymbol{\alpha}\in\Delta_q, h_i \in \mathcal{H}_{k(h_i)}}} \Bigg\{ \sum_{t=1}^T \Phi(y_t f(x_t)) + \sum_{i=1}^q \alpha_i \mathrm{Reg}_{\mathcal{H}_{k(h_i)},T}(\mathcal{A}_{k(h_i)}) + \sqrt{G^2 T \log(p)} \Bigg\}.$$

Algorithm 1 SOL.Boost

```
**Input:** Online linear learning algorithms $(\mathcal{A}_k)_{k=1}^p$ for $(\mathcal{H}_k)_{k=1}^p$, $(N_k)_{k=1}^p$ boosting stages, $\eta > 0$ learning rate, $G$ Lipschitz constant for $\Phi$.
**Initialize:** $w_{1,k} = 1, \forall k \in [1, p]$.
**for** $t = 1, \dots, T$: **do**
    **Receive:** feature $x_t$.
    **for** $k = 1, \dots, p$ **do**
        **Query:** algorithm $\mathcal{A}_k$ for hypothesis $h_{t,k}$ and prediction $h_{t,k}(x_t)$.
        **Set:** $\gamma_{t,k} = \frac{w_{t,k}}{\sum_{j=1}^p w_{t,j}}$.
    **end for**
    **Set:** predictor $h_t = \sum_{k=1}^p \gamma_{t,k} h_{t,k}$ and predict $h_t(x_t)$.
    **Receive:** label $y_t$.
    **for** $k = 1, \dots, p$ **do**
        **Attribute:** loss $l_t(y_t h_{t,k}(x_t))$ to each $\mathcal{A}_k$, where $l_t$ is the linear function $l_t \colon z \mapsto \Phi'\big(y_t h_t(x_t)\big) y_t z$.
    **end for**
    **for** $k = 1, \dots, p$ **do**
        **Update:** weight $w_{t+1,k} = w_{t,k}\,\big(1 - \eta\, l_t\big(y_t h_{t,k}(x_t)\big)\big)$
    **end for**
**end for**
```

Proof. Let $\boldsymbol{\alpha}^* \in \Delta_q$ and $(h_i^*)_{i=1}^q \subset \mathcal{H}$ be such that $\sum_{i=1}^q \alpha_i^* h_i^* \in \operatorname{conv}(\mathcal{H})$. For any $i \in [q]$ and $k \in [1, p]$, define $h_{k,i}^* = 1_{\{k(h_i^*)=k\}} h_i^*$. Then, it follows that

$$\begin{aligned}\sum_{i=1}^q \alpha_i^* h_i^* &= \sum_{i=1}^q \sum_{k=1}^p 1_{\{k(h_i^*)=k\}} \alpha_i^* h_i^* = \sum_{k=1}^p \sum_{i=1}^q 1_{\{k(h_i^*)=k\}} \alpha_i^* 1_{\{k(h_i^*)=k\}} h_i^* \\ &= \sum_{k=1}^p \left(\sum_{j=1}^q 1_{\{k(h_j^*)=k\}} \right) \left[\sum_{i=1}^q \frac{1_{\{k(h_i^*)=k\}} \alpha_i^*}{\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}} h_{k,i}^* \right] = \sum_{k=1}^p \gamma_k^* \sum_{j=1}^q \beta_{k,j}^* h_{k,i}^*,\end{aligned}$$

where $h_{k,i}^* = 1_{\{k(h_i^*)=k\}} h_i^*$, $\gamma_k^* = \sum_{j=1}^q 1_{\{k(h_j^*)=k\}}$, $\beta_{k,j}^* = \sum_{i=1}^q \frac{1_{\{k(h_i^*)=k\}} \alpha_i^*}{\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}}$. By convexity of the loss function Φ, we can write

$$\begin{aligned}&\sum_{t=1}^T \Phi\left(y_t h_t(x_t)\right) - \Phi\left(y_t \sum_{k=1}^p \gamma_k^* \sum_{j=1}^q \beta_{k,j}^* h_{k,j}^*(x_t) \right) \\ &\leq \sum_{t=1}^T \sum_{k=1}^p \Phi'\left(y_t h_t(x_t)\right) y_t h_{t,k}(x_t) \left[\gamma_{t,k} - \gamma_k^*\right] \\ &\quad + \sum_{k=1}^p \gamma_k^* \sum_{t=1}^T \Phi'\left(y_t h_t(x_t)\right) y_t \left[h_{t,k}(x_t) - \sum_{j=1}^q \beta_{k,j}^* h_{k,j}^*(x_t) \right].\end{aligned}$$

The first term in the last expression is bounded because the algorithm applies the Prod(η) algorithm [7] to the *hallucinated losses* above. Specifically, we can use the potential function $P(w_t) = \log\left(\sum_{k=1}^{p} w_{t,k}\right)$ to track the algorithm's progress against these surrogate losses and compute:

$$\begin{aligned}
\log\left(\frac{\sum_{k=1}^{p} w_{t+1,k}}{\sum_{j=1}^{p} w_{1,j}}\right) &= \log\left(\prod_{s=1}^{t} \frac{\sum_{k=1}^{p} w_{s+1,k}}{\sum_{j=1}^{p} w_{s,j}}\right) = \sum_{s=1}^{t} \log\left(\frac{\sum_{k=1}^{p} w_{s+1,k}}{\sum_{j=1}^{p} w_{s,j}}\right) \\
&= \sum_{s=1}^{t} \log\left(1 - \eta \sum_{k=1}^{p} \gamma_{s,k} \Phi'\left(y_t h_t(x_t)\right) y_t h_{t,k}(x_t)\right) \\
&\leq \sum_{s=1}^{t} -\eta \sum_{k=1}^{p} \gamma_{s,k} \Phi'\left(y_t h_t(x_t)\right) y_t h_{t,k}(x_t),
\end{aligned}$$

using the inequality $\log(1+x) \leq x$ for $x \geq -\frac{1}{2}$. We can also write, for any $k \in [1, p]$,

$$\begin{aligned}
\log\left(\frac{\sum_{i=1}^{p} w_{t+1,i}}{\sum_{j=1}^{p} w_{1,j}}\right) &\geq \log\left(\frac{w_{t+1,k}}{\sum_{j=1}^{p} w_{1,j}}\right) = -\log\left(\sum_{j=1}^{p} w_{1,j}\right) + \log\left(w_{t+1,k}\right) \\
&\geq -\log\left(\sum_{j=1}^{p} w_{1,j}\right) - \sum_{s=1}^{t} \eta \Phi'\left(y_s h_s(x_s)\right) y_s h_{s,k}(x_s) \\
&\quad - \sum_{s=1}^{t} \left(\eta \Phi'\left(y_s h_s(x_s)\right) y_s h_{s,k}(x_s)\right)^2,
\end{aligned}$$

in view of $\log(1+x) \geq x - x^2$ for all $x \geq -\frac{1}{2}$, and the constraint on η. By concavity of the logarithm, this implies that for the $\boldsymbol{\gamma}^* \in \Delta_p$ chosen above,

$$\begin{aligned}
\log\left(\frac{\sum_{i=1}^{p} w_{t+1,i}}{\sum_{j=1}^{p} w_{1,j}}\right) &\geq \log\left(\frac{\sum_{k=1}^{p} \gamma_k^* w_{t+1,k}}{\sum_{j=1}^{p} w_{1,j}}\right) \geq \sum_{k=1}^{p} \gamma_k^* \log\left(\frac{w_{t+1,k}}{\sum_{j=1}^{p} w_{1,j}}\right) \\
&\geq \sum_{k=1}^{p} \gamma_k^* \Bigg[-\log\left(\sum_{j=1}^{p} w_{1,j}\right) + \sum_{s=1}^{t} -\eta \Phi'\left(y_s h_s(x_s)\right) y_s h_{s,k}(x_s) \\
&\quad - \left(\eta \Phi'\left(y_s h_s(x_s)\right) y_s h_{s,k}(x_s)\right)^2 \Bigg].
\end{aligned}$$

Combining these calculations yields the inequality:

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{k=1}^{p} \Phi'\Big(y_t h_t(x_t)\Big) y_s h_{t,k}(x_t) (\gamma_{t,k} - \gamma_k^*) \\
&\qquad \leq \sum_{t=1}^{T} \sum_{k=1}^{p} \gamma_k^* \eta \Big(\Phi'\Big(y_t h_t(x_t)\Big) y_s h_{t,k}(x_t)\Big)^2 + \frac{1}{\eta} \log(p),
\end{aligned}$$

since $w_{1,k} = 1 \ \forall k \in [1, p]$.

For the second term, notice that if for each $k \in [1, p]$ we attribute the loss $l_t(y_t h_{t,k}(x_t))$ to $\mathcal{A}_k$, where l_t is the linear function $l_t \colon z \mapsto \Phi'\big(y_t h_t(x_t)\big) y_t z$, then the fact that l_t is linear implies that $\mathcal{A}_k$ attains some sublinear regret $\mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)$ against $\mathcal{H}_k$: $\max_{h_k^* \in \mathcal{H}_k} \sum_{t=1}^T l_t(h_{t,k}(x_t)) - \sum_{t=1}^T l_t(h_k^*(x_t)) \leq \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)$. Since l_t is a linear loss, this directly implies that the regret guarantee $\mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)$ extends to the convex hull of $\mathcal{H}_k$: $\max\limits_{h_k^* \in \mathrm{conv}(\mathcal{H}_k)} \sum_{t=1}^T l_t(h_{t,k}(x_t)) - \sum_{t=1}^T l_t(h_k^*(x_t)) \leq \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)$. It now follows that:

$$
\begin{aligned}
&\sum_{t=1}^T \Phi\Big(y_t h_t(x_t)\Big) - \Phi\Big(y_t \sum_{k=1}^p \gamma_k^* \sum_{j=1}^q \beta_{k,j}^* h_{k,j}^*(x_t)\Big) \\
&\leq \sum_{k=1}^p \gamma_k^* \eta \sum_{t=1}^T \Big(\Phi'\Big(y_t h_t(x_t)\Big) y_s h_{t,k}(x_t)\Big)^2 + \frac{1}{\eta} \log(p) + \sum_{k=1}^p \gamma_k^* \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k).
\end{aligned}
$$

Moreover, we can rewrite the convex combination of the regret quantities in terms of the original convex combination weights of the comparator hypothesis:

$$
\begin{aligned}
&\sum_{k=1}^p \gamma_k^* \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k) = \sum_{k=1}^p \gamma_k^* \sum_{i=1}^q \beta_{k,i}^* \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k) \\
&= \sum_{k=1}^p \left(\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}\right) \left[\sum_{i=1}^q \frac{1_{\{k(h_i^*)=k\}} \alpha_i^*}{\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}} \mathrm{Reg}_{\mathcal{H}_k,T}(\mathcal{A}_k)\right] \\
&= \sum_{k=1}^p \left(\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}\right) \left[\sum_{i=1}^q \frac{1_{\{k(h_i^*)=k\}} \alpha_i^*}{\sum_{j=1}^q 1_{\{k(h_j^*)=k\}}} \mathrm{Reg}_{\mathcal{H}_{k(h_i^*)},T}(\mathcal{A}_{k(h_i^*)})\right] \\
&= \sum_{k=1}^p \sum_{i=1}^q 1_{\{k(h_i^*)=k\}} \alpha_i^* \mathrm{Reg}_{\mathcal{H}_{k(h_i^*)},T}(\mathcal{A}_{k(h_i^*)}) = \sum_{i=1}^q \alpha_i^* \mathrm{Reg}_{\mathcal{H}_{k(h_i^*)},T}(\mathcal{A}_{k(h_i^*)}).
\end{aligned}
$$

Choosing $\eta = \sqrt{\frac{\log(p)}{GT}}$ satisfies the conditions and yields the desired result. □

One remarkable aspect of SOL.Boost is that it does not require knowledge of $\mathfrak{R}_T^{\mathrm{seq}}(\mathcal{H}_k)$ for any $\mathcal{H}_k$. As can be seen in Theorem 3, these complexity terms are replaced by the regret of each algorithm and are attained automatically. This is a significant advantage over the structural ensemble algorithms in the batch setting (e.g. DeepBoost [8]), which require the learner to either compute or estimate these quantities.

Moreover, the bound accompanied by SOL.Boost can vastly improve upon bounds that ignore the structural decomposition. The former realizes an average of all the regrets, and the latter is based upon the maximum regret among all base algorithms.

SOL.Boost updates all p base algorithms at each step. To improve the per-round computational cost, we can sample and query only a single base algorithm

Algorithm 2 Structural OTB

```
Input: Online algorithms $(\mathcal{A}_k)_{k=1}^p$ for decomposition $\mathcal{H} = \cup_{k=1}^p \mathcal{H}_k$, $\mathcal{J}$ family of contiguous intervals in $[1, T]$
for $t = 1, \ldots, T$: do
    Receive: feature $x_t$
    Query: algorithm $\mathcal{A}_k$ for hypothesis $h_{t,k}$
    Receive: label $y_t$ and losses $L(h_{t,k}(x_t), y_t)$ for $k \in [1, p]$
    Set: $k_t = \operatorname{argmin}_{k \in [1,p]} L(h_{t,k}(x_t), y_t)$
end for
Set: $J_{\text{out}} = \operatorname{argmin}_{J \in \mathcal{J}} \frac{1}{|J|} \sum_{t \in J} L(h_{t,k_t}(x_t), y_t) + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J|^{1/2}}}$
Output: $h_{J_{\text{out}}} = \frac{1}{|J_{\text{out}}|} \sum_{t \in J_{\text{out}}} h_{t,k_t}$
```

using techniques from the bandit literature (see e.g. [6]). This will come at the price of an extra $\sqrt{p}$ factor in the last term on the right-hand side of Theorem 3.

While it might be tempting to compare SOL.Boost to the work of [3], the algorithm presented here actually solves a regression problem, and it is more proper to compare it to the work of [2]. In fact, this is why we build upon the concept of online linear-learnability introduced in the latter paper.

The work of [12] may also seem related to the problem we consider, but it is in fact quite different since the hypothesis sets for online multiple kernel learning and online ensemble learning are distinct. Moreover, the guarantees procured there do not admit arbitrary structural decompositions as in Theorem 3 and are only in terms of the best base algorithm.

5 Online-to-batch conversion

In this section, we design an effective online-to-batch conversion technique for structural learning. Here, we assume that the learner receives a sample $((x_1, y_1), \ldots, (x_T, y_T))_{t=1}^T$ in $\mathcal{X} \times \mathcal{Y}$ drawn i.i.d. according to a distribution $\mathcal{D}$. The objective is to determine a hypothesis h based on a sequence of hypotheses $h_1, \ldots, h_T$ output by an online learning algorithm that admits a favorable generalization error $R(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x_t, y_t)]$.

We show that one can design algorithms whose generalization bounds account for the structure of the hypothesis set. Moreover, these are the first known structural generalization bounds in the batch setting that are in terms of the best-in-class hypothesis and can be viewed as an estimation error extension of the theoretical bounds in [8].

We consider a single static loss function $L \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and assume that the difference between rounds is simply the label: $L_t(h_t(x_t)) = L(h_t(x_t), y_t)$. For convenience, we also assume that L is convex in its first argument. Note that any convex surrogate used in Section 4 will satisfy this. The online-to-batch conversion technique that we present is data-dependent and can be viewed as a structural extension of the method of [9]. At a high level, finding the subpath with the smallest cumulative loss will ensure small empirical estimation error, and penalizing shorter paths will guarantee that the output will generalize well. This is not immediately obvious nor necessarily intuitive, since greedy optimization of empirical error without regularization does not lead to good generalization in many cases.

Theorem 4 presents the guarantee of our method. Figure 2 illustrates our algorithm, which benefits from the following guarantee.

Theorem 4. *Let $((x_1, y_1), \dots, (x_T, y_T))$ be an i.i.d. sample drawn from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Assume that the loss function $L \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is bounded by a constant C and that each base online algorithm in the input of Structural OTB admits the following regret guarantee: for any $k \in [1, p]$, $h_k^* \in \mathcal{H}_k$, and contiguous subset $J \subset [1, T]$: $\sum_{t \in J} L(h_{t,k}(x_t), y_t) - L(h^*(x_t), y_t) \leq \mathrm{Reg}_J(\mathcal{A}_k)$. Then, with probability at least $1 - \delta$, each of the following guarantees holds for Structural OTB:*

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[L(h_{J_{out}}(x), y)\right] \leq \min_{J^* \in \mathcal{J}, \alpha^* \in \Delta_p, h_k^* \in \mathcal{H}_k} \Bigg\{ &\sum_{k=1}^{p} \alpha_k^* \frac{1}{|J^*|} \sum_{t \in J^*} L(h_k^*(x_t), y_t) \\
&+ \sum_{k=1}^{p} \alpha_k^* \mathrm{Reg}_{J^*}(\mathcal{A}_k) + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J^*|}} \Bigg\},
\end{aligned}$$

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[L(h_{J_{out}}(x), y)\right] \leq \min_{J^* \in \mathcal{J}, \alpha^* \in \Delta_p, h_k^* \in \mathcal{H}_k} \Bigg\{ &\sum_{k=1}^{p} \alpha_k^* \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[L(h_k^*(x), y)\right] \\
&+ \sum_{k=1}^{p} \alpha_k^* \mathrm{Reg}_{J^*}(\mathcal{A}_k) + \sqrt{\frac{2C^2 \log(p/\delta)}{T}} + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J^*|}} \Bigg\}.
\end{aligned}$$

Proof. Let $J \subset [1, T]$ be any subset, and denote $h_J = \frac{1}{|J|} \sum_{t \in J} h_{t,k_t}$. Notice that by convexity of L in the first coordinate, $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(h_J(x), y)\right] \leq \frac{1}{|J|} \sum_{t \in J} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(h_{t,k_t}(x), y)\right]$.

Now for any $t \in J$, define $M_t = \frac{1}{|J|}\left(L(h_{t,k_t}(\cdot), \cdot) - \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(h_{t,k_t}(x), y)\right]\right)$. By design, $(M_t)_{t \in J}$ is a sequence of martingale differences over J, such that if we reindex $J = [1, T_J]$, then $\mathbb{E}[M_s | M_1, \dots, M_{s-1}] = 0$ for every $s \in [1, T_J]$.

Furthermore, by Azuma's inequality, we can guarantee that with probability at least $1 - \delta$, $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(h_{t,k_t}(x), y)\right] \leq \frac{1}{|J|} \sum_{t \in J} L(h_{t,k_t}(x_t), y_t) + \sqrt{\frac{2C^2 \log(1/\delta)}{|J|}}$. By applying a union bound over all $J \in \mathcal{J}$, then with probability at least $1 - \delta$, the following bound holds for every $J \in \mathcal{J}$:

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[L(h_{t,k_t}(x), y)\right] \leq \frac{1}{|J|} \sum_{t \in J} L(h_{t,k_t}(x_t), y_t) + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J|}}.$$

Thus, it follows that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[L(h_{J_{\mathrm{out}}}(x), y)\right] \leq \min_{J^* \in \mathcal{J}} \frac{1}{|J^*|} \sum_{t \in J^*} L(h_{t,k_t}(x_t), y_t) + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J^*|}}.$$

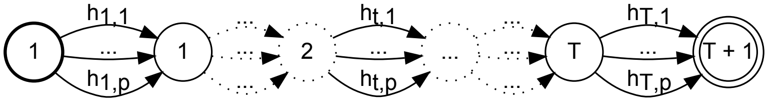

Fig. 2. Illustration of Structural OTB. The algorithm chooses the best a posteriori sub-path of hypotheses among all algorithms.

By the choice of k_t, we can further say that $L(h_{t,k_t}(x_t), y_t) \leq L(h_{t,k}(x_t), y_t)$ $\forall k \in [1, p]$. In particular, this means that for any $J \in \mathcal{J}$,

$$\frac{1}{|J|} \sum_{t \in J} L(h_{t,k_t}(x_t), y_t) \leq \min_{\alpha^* \in \Delta_p} \frac{1}{|J|} \sum_{t \in J} \sum_{k=1}^{p} \alpha_k^* L(h_{t,k}(x_t), y_t)$$

$$\leq \min_{\alpha^* \in \Delta_p, h_k^* \in \mathcal{H}_k} \sum_{k=1}^{p} \alpha_k^* \left(\frac{1}{|J|} \sum_{t \in J} L(h_k^*(x_t), y_t) + \mathrm{Reg}_J(\mathcal{A}_k) \right).$$

Combining the above two inequalities yields the first result.

Furthermore, we can use Hoeffding's inequality over the best-in-class classifier's guarantee for each of the p subclasses and apply a union bound to say that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [L(h_{J_{\mathrm{out}}}(x), y)] \leq \min_{J^* \in \mathcal{J}, \alpha^* \in \Delta_p, h_k^* \in \mathcal{H}_k} \left\{ \sum_{k=1}^{p} \alpha_k^* \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [L(h_k^*(x), y)] \right.$$

$$\left. + \sum_{k=1}^{p} \alpha_k^* \mathrm{Reg}_{J^*}(\mathcal{A}_k) + \sqrt{\frac{2C^2 \log(p/\delta)}{T}} + \sqrt{\frac{2C^2 \log(|\mathcal{J}|/\delta)}{|J^*|}} \right\}. \qquad \square$$

Note that one natural choice of $\mathcal{J}$, as discussed by [9], is the set of all suffixes of $[1, T]$: $\{[1, T], [2, T], \ldots, [T, T]\}$. This was shown empirically to outperform the "data-independent" online-to-batch conversion methods of [5]. With this specific choice of $\mathcal{J}$, $|\mathcal{J}| = T$, and the logarithmic dependence on $|\mathcal{J}|$ is mild.

6 Conclusion

We presented a series of theoretical and algorithmic results for structural online learning. Our theory and algorithms can be further extended to cover other learning settings, including multi-class classification, regression and general online learning. In contrast with the batch algorithms for structural learning, our algorithms do not require the estimation of the Rademacher complexities in the decomposition of the hypothesis set. Moreover, our online-to-batch conversion algorithm provides an efficient alternative to the current structural ensemble methods used in the batch setting.

Acknowledgements. This work was partly funded by the NSF awards IIS-1117591 and CCF-1535987 and was also supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1342536.

Bibliography

1. Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, 3, 2002.
2. Alina Beygelzimer, Elad Hazan, Satyen Kale, and Haipeng Luo. Online gradient boosting. In *Proceedings of NIPS*, pages 2449–2457, 2015.
3. Alina Beygelzimer, Satyen Kale, and Haipeng Luo. Optimal and adaptive algorithms for online boosting. In *ICML*, volume 37 of *JMLR Proceedings*, 2015.
4. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
5. Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.
6. Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.
7. Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.
8. Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *Proceedings of ICML*, 2014.
9. Ofer Dekel and Yoram Singer. Data-driven online to batch conversions. In *NIPS*, pages 267–274, 2005.
10. Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.
11. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer System Sciences*, 55(1):119–139, 1997.
12. Rong Jin, Steven CH Hoi, and Tianbao Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *International Conference on Algorithmic Learning Theory*, pages 390–404. Springer, 2010.
13. Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, pages 1–50, 2002.
14. Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.
15. Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize : From value to algorithms. In *NIPS*, pages 2150–2158, 2012.
16. Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In *Proceedings of NIPS*, pages 1984–1992, 2010.
17. Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.