A. General path expert graphs

For simplicity, we presented our algorithms and guarantees in the case of the graph G admitting all path experts. In most cases in practice, different acyclic graph of experts such as that of Figure 2 must be considered. This occurs in particular because of the presence of known constraints restricting the set of admissible sequences of substructures.

For example, the learning problem may consist of predicting the pronunciation associated to each sequence of words. In that case, for most languages, there exist phonotactic rules making some phonemic sequences inadmissible. Similarly, in parsing or translation tasks, some word sequences can be ruled out because they do not conform to some clear syntactic or stylistic rule. Let A denote a finite automaton accepting admissible sequences of $\mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$ and let G_t denote the graph of path experts considered at round $t \in [1, T]$, with $G_1 = G$. At each round $t \in [1, T]$, the learner receives a new input sequence $\mathbf{x}_t$ that is used to derive a finite automaton $\widetilde{G}_t$ from G_t by replacing in G_t the substructure predictor h_j^k, $j \in [1, p]$, $k \in [1, l]$, by its prediction $h_j^k(\mathbf{x}_t)$. Since some sequences of $\widetilde{G}_t$ may not be admissible, we must remove from G_t path experts generating sequences not in $\widetilde{G}_t \cap A$. This can be achieved straightforwardly using the intersection algorithm for finite automata if we keep track, for each substructure predicted, of the original substructure expert generating that prediction. G_{t+1} is the resulting graph of admissible path experts.

The on-line learning algorithm we consider, WMWP, applies to an arbitrary directed acyclic graph and thus can be applied to graph G_t at each round t. The distribution over the path experts maintained by WMWP effectively assigns probability zero to the path experts not present in G_t at round t. Our learning guarantees hold for this more general setting and in fact end up being more favorable since the cardinality of the set of admissible path experts is smaller than that of graph G.

B. On-line-to-batch conversion

Lemma 6. *For any $t \in [1, T]$, the following identity holds:*

$$\sum_{\mathsf{h} \in \mathsf{H}} \mathsf{p}_t(\mathsf{h}) L(\mathsf{h}(\mathbf{x}_t), \mathbf{y}_t)) = \sum_{k=1}^{l} \sum_{j=1}^{p} w_{t,kj} \ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k).$$

Proof. Recall that for any $t \in [1, T]$ and $k \in [1, l]$, $\sum_{j=1}^{p} w_{t,kj} = 1$. Thus, let $w_{t,k\cdot}$ denote the distribution defined by the non-negative weights $w_{t,kj}$. Then, the following chain of equalities proves the result:

$$\begin{aligned}
\sum_{\mathsf{h} \in \mathsf{H}} \mathsf{p}_t(\mathsf{h}) L(\mathsf{h}(\mathbf{x}_t), \mathbf{y}_t)) &= \mathop{\mathbb{E}}_{\mathsf{h} \sim \mathsf{p}_t} [L(\mathsf{h}(\mathbf{x}_t), \mathbf{y}_t)] \\
&= \mathop{\mathbb{E}}_{\mathsf{h} \sim \mathsf{p}_t} [\sum_{k=1}^{l} \ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k)] \\
&= \sum_{k=1}^{l} \mathop{\mathbb{E}}_{\mathsf{h} \sim \mathsf{p}_t} [\ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k)] \\
&= \sum_{k=1}^{l} \mathop{\mathbb{E}}_{\substack{\mathsf{h}^1 \sim w_{t,1\cdot} \\ \cdots \\ \mathsf{h}^l \sim w_{t,l\cdot}}} [\ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k)] \\
&= \sum_{k=1}^{l} \mathop{\mathbb{E}}_{\mathsf{h}^k \sim w_{t,k\cdot}} [\ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k)] \\
&= \sum_{k=1}^{l} \sum_{j=1}^{p} w_{t,kj} \ell_k(\mathsf{h}^k(\mathbf{x}_t), y_t^k).
\end{aligned}$$

□

Proposition 4. *The following bound holds for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$:*

$$\begin{aligned}
\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{MVote}}(\mathbf{x}), \mathbf{y})] \leq 2\, \mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{Rand}}(\mathbf{x}), \mathbf{y})] \\
- 2\, \mathbb{E}[\gamma(\mathbf{x}, \mathbf{y})],
\end{aligned}$$

where $\gamma(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{l} \gamma_k(\mathbf{x}, \mathbf{y})$ with

$$\gamma_k(\mathbf{x}, \mathbf{y}) = \max\left(0, \frac{1}{|\mathcal{P}_\delta|} \sum_{\mathsf{p}_t \in \mathcal{P}_\delta} \sum_{j=1}^{p} w_{t,kj} \mathbf{1}_{h_j^k(\mathbf{x}) \neq y^k} - \frac{1}{2}\right).$$

Proof. The proof is a slight refinement of that of Proposition 3. If $\mathcal{H}_{\text{MVote}}$ makes an error at position k on example $(\mathbf{x}, \mathbf{y})$ then the total weight of incorrect labels at that position must be $\frac{1}{2} + \gamma_k(\mathbf{x}, \mathbf{y})$. In other words, we have the following inequality

$$\mathbf{1}_{\mathcal{H}_{\text{MVote}}^k(\mathbf{x}) \neq y^k} \leq \frac{2}{|\mathcal{P}_\delta|} \sum_{p \in \mathcal{P}_\delta} \sum_{j=1}^{p} w_{t,kj} \mathbf{1}_{h_j^k(\mathbf{x}) \neq y^k} - 2\gamma_k(\mathbf{x}, \mathbf{y})$$

when $\mathbf{1}_{\mathcal{H}_{\text{MVote}}^k(\mathbf{x}) \neq y^k} = 1$. Since the right-hand side of the bound above is always positive, it also holds when $\mathbf{1}_{\mathcal{H}_{\text{MVote}}^k(\mathbf{x}) \neq y^k} = 0$. The rest of the proof is the same as that of Proposition 3. □

Theorem 2. *For any $\delta > 0$, with probability at least $1 - \delta$ over the choice of the sample $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T))$ drawn i.i.d. according to $\mathcal{D}$, the following inequalities*

hold:

$$\mathbb{E}[L(\mathcal{H}_{\text{Rand}}(\mathbf{x}),\mathbf{y})] \leq \inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] + \frac{R_T}{T} + 2M\sqrt{\frac{\log\frac{2}{\delta}}{T}}$$

$$\mathbb{E}[L(\mathcal{H}_{\text{Rand}}(\mathbf{x}),\mathbf{y})] \leq \inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] + 2M\sqrt{\frac{l\log p}{T}} + 2M\sqrt{\frac{\log\frac{2}{\delta}}{T}}.$$

Proof. Since there are only finitely many expert paths h, there is an expert path $\mathsf{h}^* \in \mathsf{H}$ such that $\inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] = \mathbb{E}[L(\mathsf{h}^*(\mathbf{x}),\mathbf{y})]$. By Hoeffding's inequality, the probability of the event

$$\left\{\frac{1}{T}\sum_{t=1}^{T} L(\mathsf{h}^*(\mathbf{x}_t),\mathbf{y}_t) - \mathbb{E}[L(\mathsf{h}^*(\mathbf{x}),\mathbf{y})] > M\sqrt{\frac{\log\frac{2}{\delta}}{T}}\right\}$$

is at most $\delta/2$. Therefore, by Proposition 1 and the union bound, the following holds with probability at least $1-\delta$:

$$\begin{aligned}
&\mathbb{E}[L(\mathcal{H}_{\text{Rand}}(\mathbf{x}),\mathbf{y})] - \inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] \\
&\leq \frac{1}{T}\sum_{t=1}^{T} L_t + M\sqrt{\frac{\log\frac{2}{\delta}}{T}} - \frac{1}{T}\sum_{t=1}^{T} L(\mathsf{h}^*(\mathbf{x}_t),\mathbf{y}_t) + M\sqrt{\frac{\log\frac{2}{\delta}}{T}} \\
&\leq \frac{R_T}{T} + 2M\sqrt{\frac{\log\frac{2}{\delta}}{T}},
\end{aligned}$$

which proves the first inequality. The regret of the randomized MW algorithm for losses taking values in $[0,1]$ is known to be bounded by $2\sqrt{T\log N}$ where N is the number of experts (Cesa-Bianchi & Lugosi, 2006). In our context, this gives $R_T \leq 2M\sqrt{T\log(p^l)}$. Plugging in this bound in the first inequality of the theorem yields directly the second one. □

C. Cross-validation based on-line-to-batch conversion

Cesa-Bianchi et al. (2004) described an on-line-to-batch conversion technique based on a cross-validation approach. Given a sequence of hypotheses produced by an on-line algorithm, a single hypothesis is selected based on its empirical loss on unseen examples plus a special penalty term. These results can be easily generalized to the case where an on-line algorithm produces distributions over hypotheses rather than just a single hypothesis. More precisely, suppose that an on-line algorithm generates a sequence of distributions $\mathsf{p}_1,\ldots,\mathsf{p}_T$ over some finite set of hypotheses H. We define

$$\Theta(\mathsf{p}_t) = \frac{1}{T-t}\sum_{s=t+1}^{T} L_s(\mathsf{p}_t), \tag{11}$$

where $L_s(\mathsf{p}_t) = \sum_{\mathsf{h}\in\mathsf{H}} \mathsf{p}_t(\mathsf{h}) L(\mathsf{h}(\mathbf{x}_s),\mathbf{y}_s)$ and L is a given loss function bounded by M. We also set $c_\delta(s) = \sqrt{\frac{1}{2s}\log\frac{T(T+1)}{\delta}}$. Define

$$\widehat{\mathsf{p}} = \operatorname*{argmin}_{\mathsf{p}_t} (\Theta(\mathsf{p}_t) + c_\delta(T-t)). \tag{12}$$

If $\mathcal{H}_{\text{CVRand}}$ is a randomized hypothesis that, given example $\mathbf{x}$, first chooses $\mathsf{h} \in \mathsf{H}$ according to $\widehat{\mathsf{p}}$ and predicts $\mathsf{h}(\mathbf{x})$, then the following result holds.

Theorem 7. *For $\widehat{\mathsf{p}}$ and $\mathcal{H}_{\text{CVRand}}$ defined as above, with probability at least $1-\delta$ the following inequality holds:*

$$\mathbb{E}[L(\mathcal{H}_{\text{CVRand}}(\mathbf{x}),\mathbf{y})] \geq \frac{1}{T}\sum_{t=1}^{T} L_t(\mathsf{p}_t) + 6\sqrt{\frac{1}{T}\log\frac{2(T+1)}{\delta}}.$$

The proof of this result is identical to the proof of Theorem 4 in (Cesa-Bianchi et al., 2004). This result leads us to introduce an alternative ensemble structured prediction algorithm: first we use WMWP as in Section 3 to generate a sequence of distributions $\mathsf{p}_1,\ldots,\mathsf{p}_T$ over path experts in H; next a single distribution $\widehat{\mathsf{p}}$ is chosen to minimize (12). As discussed in Section 3, $\widehat{\mathsf{p}}$ a distribution can be represented using a matrix $\widehat{\mathbf{W}} = (\widehat{w}_{kj})_{kj} \in \mathbb{R}^{l\times p}$. To make predictions we can use either the randomized hypothesis $\mathcal{H}_{\text{CVRand}}$ defined above, or the majority vote hypothesis

$$\mathcal{H}_{\text{CV}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}} \prod_{k=1}^{l}\left(\sum_{j=1}^{p} \widehat{w}_{kj}\mathbf{1}_{h_j^k(\mathbf{x})=y^k}\right). \tag{13}$$

Theorem 7 combined with Hoeffding's inequality and the regret bounds of Cesa-Bianchi & Lugosi (2006) yield the following result.

Theorem 8. *For any $\delta > 0$, with probability $1-\delta$ over the choice of the sample $((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_T,\mathbf{y}_T))$ drawn i.i.d according to $\mathcal{D}$ the following inequalities hold:*

$$\begin{aligned}
\mathbb{E}[L(\mathcal{H}_{\text{CVRand}}(\mathbf{x}),\mathbf{y})] &\leq \inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] + \frac{R_T}{T} \\
&\quad + M\sqrt{\frac{\log\frac{2}{\delta}}{T}} + 6M\sqrt{\frac{1}{T}\log\frac{4(T+1)}{\delta}} \\
\mathbb{E}[L(\mathcal{H}_{\text{CVRand}}(\mathbf{x}),\mathbf{y})] &\leq \inf_{\mathsf{h}\in\mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}),\mathbf{y})] + 2M\sqrt{\frac{l\log p}{T}} \\
&\quad + M\sqrt{\frac{\log\frac{2}{\delta}}{T}} + 6M\sqrt{\frac{1}{T}\log\frac{4(T+1)}{\delta}}.
\end{aligned}$$

The learning guarantees for $\mathcal{H}_{\text{CV}}$ can now be derived using either Proposition 3 or Proposition 4.

Algorithm 3 Follow the Perturbed Leader, FLP.

Inputs: set of experts $\{h_1, \ldots, h_p\}$; sample $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T)\}$; parameter $\epsilon \in (0, \frac{1}{pl}]$;
for $t = 1$ **to** T **do**
  for $k = 1$ **to** l **do**
    sample $\mathbf{q} = (q_1^k, \ldots, q_p^k)$ with density $\propto e^{-\epsilon \|\mathbf{q}\|_1}$;
    $\mathsf{h}_t^k \leftarrow \operatorname{argmin}_{h_j^k} \sum_{s=1}^{t-1} \ell(h_j^k(\mathbf{x}_s), \mathbf{y}_s) + q_j^k$
  end for
  $\mathsf{h}_t \leftarrow (\mathsf{h}_t^1, \ldots, \mathsf{h}_t^l)$
end for
Return $\{\mathsf{h}_1, \ldots, \mathsf{h}_T\}$

D. FPL-based algorithm

In Section 3, we presented a solution to the ensemble problem for structured prediction tasks based on the WMWP algorithm. Here, we present an alternative approach based on the FPL algorithm. The main difference with the case of the WMWP algorithm is that, at each iteration, FPL outputs a path expert h_t rather than a distribution. However, this can be viewed as producing a probability point mass p_t at h_t. Thus, the on-line-to-batch conversions we described for WMWP also apply here as well.

We first briefly describe the FPL algorithm. The idea of the algorithm is simple. At each round of the on-line algorithm, we attempt to choose the path that has been the best performer so far. However, it can be shown that this deterministic approach is suboptimal. Thus, we regularize our selection procedure by adding some random perturbation to the cumulative loss of each path before making our choice. As before, the difficulty is that keeping track of the cumulative loss of each path in the graph G is inefficient. Kalai & Vempala (2005) showed that it is sufficient to store only the cumulative losses of each edge and only add random perturbations to each edge in the graph. We remark that, for the graph G, finding the current best path is straightforward: just traverse the graph from vertex 0 to vertex l by selecting the edge with the best perturbed cumulative loss. See pseudocode for the FPL algorithm in Algorithm 3 for more details.

The output of the FPL Algorithm is a set of path experts $\{\mathsf{h}_1, \ldots, \mathsf{h}_T\}$. Next, to extract a subset $\mathcal{H} \subseteq \{\mathsf{h}_1, \ldots, \mathsf{h}_T\}$, we can use the objective function Γ of (4) where p_t is now just a point mass at h_t. Once a collection $\mathcal{H}$ is determined, we again have two different prediction rules. Given input $\mathbf{x}$, a randomized prediction rule chooses a path $\mathsf{h} \in \mathcal{H}$ uniformly at random and predicts $\mathsf{h}(\mathbf{x})$. This hypothesis is denoted by $\mathcal{H}_{\text{FPLRand}}$. The corresponding majority vote hypothesis $\mathcal{H}_{\text{FPL}}$, as the name suggests, predicts using majority vote at each position k. The following learning guarantees hold.

Theorem 9. *For any $\delta > 0$, with probability $1 - \delta$ over the choice of the sample $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T))$ drawn i.i.d according to $\mathcal{D}$, the following inequalities hold:*

$$\mathbb{E}[L(\mathcal{H}_{\text{FPLRand}}(\mathbf{x}), \mathbf{y})] \leq \inf_{\mathsf{h} \in \mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}), \mathbf{y})] + \frac{R_T}{T} + 3M\sqrt{\frac{\log \frac{3}{\delta}}{T}}$$

$$\mathbb{E}[L(\mathcal{H}_{\text{FPLRand}}(\mathbf{x}), \mathbf{y})] \leq \inf_{\mathsf{h} \in \mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}), \mathbf{y})] + \sqrt{\frac{Mpl^2 \log(pl)}{T}} + 3M\sqrt{\frac{\log \frac{3}{\delta}}{T}}.$$

This result is a direct consequence of Theorem 2 (where we use point masses for distributions p_t) and the bound on the regret R_T of FPL algorithm: $R_T \leq \sqrt{Mpl^2 \log pl}$.[4] We remark that since FPL is itself a randomized algorithm, we have to consider the expected regret

$$R_T = \mathop{\mathbb{E}}_{\mathbf{q}}\left[\sum_{t=1}^T L(\mathsf{h}_t(\mathbf{x}_t), \mathbf{y}_t)\right] - \inf_{\mathsf{h} \in \mathsf{H}} \sum_{t=1}^T L(\mathsf{h}(\mathbf{x}_t), \mathbf{y}_t), \quad (14)$$

where the subscript for the expectation sign indicates that the expectation is taken with respect to the random variables $\mathbf{q}$ used to define each h_t. Note that Azuma's inequality implies that with probability at least $1 - \delta$, the following holds:

$$\frac{1}{T}\sum_{t=1}^T L(\mathsf{h}_t(\mathbf{x}_t), \mathbf{y}_t) \leq \mathop{\mathbb{E}}_{\mathbf{q}}\left[\sum_{t=1}^T L(\mathsf{h}_t(\mathbf{x}_t), \mathbf{y}_t)\right] + M\sqrt{\frac{\log \frac{1}{\delta}}{T}}.$$

This additional approximation step is the reason for the factor of 3 instead of 2 in the last term in the bound.

The bounds of Theorem 9 should be compared to those of Theorem 2. For $M = 1$, as for the normalized Hamming loss, and $pl \geq 4$ the regret bound of Theorem 9 is more favorable. The learning guarantees for $\mathcal{H}_{\text{FPL}}$ now follow from a straightforward application of Proposition 3 or Proposition 4.

Finally, instead of using Γ to find $\mathcal{H}$, we can apply the cross-validation approach of (Cesa-Bianchi et al., 2004) to find a single path expert $\widehat{\mathsf{h}} \in \{\mathsf{h}_1, \ldots, \mathsf{h}_T\}$ and use it to make predictions. To keep our notation consistent, we set $\mathcal{H}_{\text{FPL-CV}} = \widehat{\mathsf{h}}$. An analogue of Theorem 7 can be established for $\mathcal{H}_{\text{FPL-CV}}$ using results from (Cesa-Bianchi et al., 2004) and the regret bounds of FPL algorithm (Cesa-Bianchi & Lugosi, 2006).

[4] The regret of the FPL algorithm for the equivalent on-line shortest path problem is bounded by $\sqrt{KL^*|E|\log|E|}$ (Cesa-Bianchi & Lugosi, 2006), where L^* is the loss of the best path in hindsight, $|E|$ is the number of edges in the graph, K is the bound on the length of a path from source to sink and it is assumed that $\ell \in [0, 1]$.

Theorem 10. *For any $\delta > 0$, with probability $1-\delta$ over the choice of the sample $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T))$ drawn i.i.d. according to $\mathcal{D}$ the following inequalities hold:*

$$\mathbb{E}[L(\mathcal{H}_{\text{FPL-CV}}(\mathbf{x}), \mathbf{y})] \leq \inf_{\mathsf{h} \in \mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}), \mathbf{y})] + \frac{R_T}{T} + 2M\sqrt{\frac{\log \frac{3}{\delta}}{T}} + 6M\sqrt{\frac{1}{T}\log\frac{3(T+1)}{\delta}}$$

$$\mathbb{E}[L(\mathcal{H}_{\text{FPL-CV}}(\mathbf{x}), \mathbf{y})] \leq \inf_{\mathsf{h} \in \mathsf{H}} \mathbb{E}[L(\mathsf{h}(\mathbf{x}), \mathbf{y})] + \sqrt{\frac{Mpl^2 \log pl}{T}} + 2M\sqrt{\frac{\log \frac{3}{\delta}}{T}} + 6M\sqrt{\frac{1}{T}\log\frac{3(T+1)}{\delta}}.$$

Our experimental results show, however, that using a single path expert to make all predictions yields a poor performance in practice.

E. Alternative algorithms and derandomizations

The WMWP algorithm applies to any resulting graph G' and the randomized algorithm we described can be used in a similar way. The resulting learning guarantees are then somewhat more favorable since the number of path experts in G' will be smaller. However, the computation of the deterministic majority-vote solution is less straightforward since (6) then becomes a constrained optimization. The problem consists of finding the most probable sequence in a non-deterministic weighted automaton and can be solved using a weighted determinization algorithm combined with a standard shortest-path algorithm (Mohri & Riley, 2002). But, while this is often efficient in practice, the worst case complexity is exponential. In such cases, one may resort to an approximate solution based on a Viterbi approximation by selecting the path (not the string) that is the most probable.

Other derandomization schemes are possible. For instance, one can also only partially derandomize the prediction by choosing $\mathsf{p}_t \in \mathcal{P}$ at random and then using p_t for a majority vote, or the approximate algorithm just described. However, this hybrid approach inherits the worst traits of its parents: the randomized predictions of the stochastic scheme and the less favorable learning guarantees of the majority vote (see Appendix F for a detailed analysis of the learning guarantees for this hybrid approach).

F. Partial derandomizations

In this section, we present learning guarantees for the partial derandomization scheme discussed in Appendix E. This can be described as follows: upon receiving an input $\mathbf{x}$, we draw a distribution $\mathsf{p}_t \in \mathcal{P}$ uniformly at random and predict $\mathcal{H}_{\text{MV},\,\mathsf{p}_t}(\mathbf{x})$ where $\mathcal{H}_{\text{MV},\,\mathsf{p}_t}$ denotes a majority vote hypothesis based on the distribution p_t. We denote the resulting hypothesis by $\mathcal{H}_{\text{RMV}}$.

Lemma 11. *The following inequality relates the error of the randomized and majority-vote hypotheses:*

$$\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{RMV}}(\mathbf{x}), \mathbf{y})] \leq 2\,\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{Rand}}(\mathbf{x}), \mathbf{y})],$$

where the expectations are taken both with respect to $\mathcal{D}$ and p.

Proof. By definition of $\mathcal{H}_{\text{RMV}}$, we can write

$$\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{RMV}}(\mathbf{x}), \mathbf{y})] = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{MV},\mathsf{p}_t}(\mathbf{x}), \mathbf{y})]$$

If $\mathcal{H}_{\text{R},\mathsf{p}_t}$ denotes a stochastic hypothesis based on p_t, then, by Proposition 3 we will have that

$$\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{MV},\mathsf{p}_t}(\mathbf{x}), \mathbf{y})] \leq 2\,\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{R},\mathsf{p}_t}(\mathbf{x}), \mathbf{y})].$$

Averaging over t yields

$$\begin{aligned}\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{RMV}}(\mathbf{x}), \mathbf{y})] &\leq \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{R},\mathsf{p}_t}(\mathbf{x}), \mathbf{y})] \\ &= 2\,\mathbb{E}[L_{\text{Ham}}(\mathcal{H}_{\text{Rand}}(\mathbf{x}), \mathbf{y})],\end{aligned}$$

where the last equality follows from the definition of $\mathcal{H}_{\text{Rand}}$. □

Based on this lemma we can give the same learning guarantees for $\mathcal{H}_{\text{RMV}}$ as for $\mathcal{H}_{\text{MVote}}$ in Theorem 2. However, as noted in Appendix E this hybrid approach inherits the worst traits of its parents: randomized predictions of the stochastic scheme and less favorable learning guarantees of the majority vote.

G. ESPBoost

G.1. Bound on the empirical Hamming loss

We first derive an upper bound on the empirical normalized Hamming loss of a hypothesis $\mathcal{H}_{\text{ESPBoost}}$, with $\widetilde{\mathsf{h}} = \sum_{t=1}^{T} \alpha_t \widetilde{\mathsf{h}}_t$.

Lemma 5. *The following upper bound holds for the empirical loss of the hypothesis $\mathcal{H}_{\text{ESPBoost}}$:*

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

Proof. Note that in view of (9), we can write, for any k and $\mathbf{x} \in \mathcal{X}$,

$$\mathcal{H}^k_{\text{ESPBoost}}(\mathbf{x}) = \operatorname*{argmax}_{y^k \in \mathcal{Y}_k} \widetilde{\mathsf{h}}^k(\mathbf{x}, y^k). \tag{15}$$

where $\widetilde{\mathsf{h}}^k = \sum_{t=1}^T \alpha_t \widetilde{\mathsf{h}}^k_t$ and $\widetilde{\mathsf{h}}^k_t(\mathbf{x}, y^k) = \mathbf{1}_{\mathsf{h}^k_t(\mathbf{x})=y^k}$. Observe that $\rho(\widetilde{\mathsf{h}}^k_t, \mathbf{x}_i, \mathbf{y}_i) = 1$ if the prediction made by h_t for the input $\mathbf{x}_i$ is correct at position k, and -1 otherwise. For any $i \in [1, m]$, by the sub-additivity of the max function,

$$\mathbf{1}_{\mathcal{H}^k_{\text{ESPBoost}}(\mathbf{x}_i) \neq y^k_i} = \mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k, \mathbf{x}_i, \mathbf{y}_i) \leq 0} \leq \mathbf{1}_{\sum_{t=1}^T \alpha_t \rho(\widetilde{\mathsf{h}}^k_t, \mathbf{x}_i, \mathbf{y}_i) \leq 0}.$$

Thus, the empirical loss of the hypothesis $\mathcal{H}_{\text{ESPBoost}}$, $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim S}[L_{\text{Ham}}(\mathcal{H}_{\text{ESPBoost}}(\mathbf{x}), \mathbf{y})]$, can be upper bounded as follows:

$$\begin{aligned}
&\frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \mathbf{1}_{\mathcal{H}^k_{\text{ESPBoost}}(\mathbf{x}_i) \neq y^k_i} \\
&\quad \leq \frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \mathbf{1}_{\sum_{t=1}^T \alpha_t \rho(\widetilde{\mathsf{h}}^k_t, \mathbf{x}_i, \mathbf{y}_i) \leq 0} \\
&\quad \leq \frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \exp\Big(- \sum_{t=1}^T \alpha_t \rho(\widetilde{\mathsf{h}}^k_t, \mathbf{x}_i, \mathbf{y}_i)\Big),
\end{aligned}$$

where we used for the last inequality the identity ($\mathbf{1}_{u \leq 0} \leq e^{-u}$) valid for all $u \in \mathbb{R}$. □

G.2. Coordinate descent

Here we present the details of the derivation of our coordinate descent algorithm.

Let $\boldsymbol{\alpha}_{t-1} \in \mathbb{R}^N$ denote the vector obtained after $t-1$ iterations and $\mathbf{e}_t$ the tth unit vector in $\mathbb{R}^N$. We denote by $\mathcal{D}_t$ the distribution over $[1, m] \times [1, l]$ defined by

$$\mathcal{D}_t(i,k) = \frac{\frac{1}{ml} \exp\Big(- \sum_{u=1}^{t-1} \alpha_u \rho(\widetilde{\mathsf{h}}^k_u, \mathbf{x}_i, \mathbf{y}_i)\Big)}{A_{t-1}}$$

where A_{t-1} is a normalization factor, $A_{t-1} = \frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \exp\big(- \sum_{u=1}^{t-1} \alpha_u \rho(\widetilde{\mathsf{h}}^k_u, \mathbf{x}_i, \mathbf{y}_i)\big)$. The direction $\mathbf{e}_t$ selected at the tth round is the one minimizing the directional derivative, that is

$$\begin{aligned}
\frac{dF(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_t)}{d\eta}\bigg|_{\eta=0} &= - \sum_{i=1}^m \sum_{k=1}^l \rho(\widetilde{\mathsf{h}}^k_t, \mathbf{x}_i, \mathbf{y}_i) \mathcal{D}_t(i,k) A_{t-1} \\
&= \big[2 \sum_{i,k:\mathsf{h}^k_t(\mathbf{x}_i) \neq y^k_i} \mathcal{D}_t(i,k) - 1\big] A_{t-1} \\
&= (2\epsilon_t - 1) A_{t-1},
\end{aligned}$$

where ϵ_t is the average error of h_t given by

$$\epsilon_t = \sum_{i=1}^m \sum_{k=1}^l \mathcal{D}_t(i,k) \mathbf{1}_{\mathsf{h}^k_t(\mathbf{x}_i) \neq y^k_i} = \mathop{\mathbb{E}}_{(i,k)\sim \mathcal{D}_t} [\mathbf{1}_{\mathsf{h}^k_t(\mathbf{x}_i) \neq y^k_i}].$$

The remaining steps of our algorithm can be determined as in the case of AdaBoost. In particular, given the direction $\mathbf{e}_t$, the best step α_t is obtained by solving the equation $\frac{dF(\boldsymbol{\alpha}_{t-1} + \alpha_t \mathbf{e}_t)}{d\alpha_t} = 0$, which admits the closed-form solution $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$. The distribution $\mathcal{D}_{t+1}$ can be expressed in terms of $\mathcal{D}_t$ with the normalization factor $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

G.3. Learning guarantees

This section presents both a margin-based generalization bound in support of the ESPBoost algorithm, and a bound on the empirical margin loss.

For any $\rho > 0$, we define the empirical margin loss of $\mathcal{H}_{\text{ESPBoost}}$ by the following:

$$\widehat{R}_\rho\bigg(\frac{\widetilde{\mathsf{h}}}{\|\boldsymbol{\alpha}\|_1}\bigg) = \frac{1}{ml} \sum_{i=1}^m \sum_{k=1}^l \mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k, \mathbf{x}_i, \mathbf{y}_i) \leq \rho \|\boldsymbol{\alpha}\|_1}. \tag{16}$$

where $\widetilde{\mathsf{h}}$ is the corresponding scoring function.

Theorem 12. *Let $\mathcal{F}$ denote the set of functions $\mathcal{H}_{\text{ESPBoost}}$ 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 $\mathcal{H}_{\text{ESPBoost}} \in \mathcal{F}$:*

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

where $\mathfrak{R}_m(H^k)$ denotes the Rademacher complexity of the class of functions

$$H^k = \{\mathbf{x} \mapsto \mathbf{1}_{h^k_j(\mathbf{x}) = y} : j \in [1, p], y \in \mathcal{Y}_k\}.$$

Proof. By definition of the Hamming loss, we can write

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} [L_{\text{Ham}}(\mathcal{H}_{\text{ESPBoost}}(\mathbf{x}), \mathbf{y})] \\
&\quad = \frac{1}{l} \sum_{k=1}^l \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} [\mathbf{1}_{\mathcal{H}^k_{\text{ESPBoost}}(\mathbf{x}) \neq \mathbf{y}}] \\
&\quad = \frac{1}{l} \sum_{k=1}^l \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} [\mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k, \mathbf{x}, \mathbf{y}) \leq 0}].
\end{aligned}$$

We bound each of the summands above separately. Let $\Pi(H^k)$ denote the convex hull of H^k. Then, for any $k \in [1, l]$, we can apply a multi-class classification bound based on the Rademacher complexity of $\Pi(H^k)$ (Koltchinskii & Panchenko, 2002; Mohri et al., 2012). Thus, for any

$\delta > 0$, with probability at least $1-\delta$, the following inequality holds:

$$\mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k,\mathbf{x},\mathbf{y})\leq 0}] \leq \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim S}[\mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k,\mathbf{x},\mathbf{y})\leq\rho\|\boldsymbol{\alpha}\|_1}] + \frac{2|\mathcal{Y}_k|^2}{\rho}\mathfrak{R}_m(\Pi(H^k)) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

Since the Rademacher complexity of the convex hull of a set coincides with that of the set, for any k, $\mathfrak{R}_m(\Pi(H^k)) = \mathfrak{R}_m(H^k)$. Thus, by the union bound, summing up over k these inequalities and dividing by l yields that for any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $\mathcal{H}_{\text{ESPBoost}} \in \mathcal{F}$:

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

which concludes the proof. □

Thus, the 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 for ESPBoost.

Theorem 13. *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*

$$\widehat{R}_\rho\left(\frac{\widetilde{\mathsf{h}}}{\|\boldsymbol{\alpha}\|_1}\right) \leq 2^T\prod_{t=1}^{T}\sqrt{\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}.$$

Proof. The proof steps are the same as those used for the bound on the empirical margin loss for AdaBoost (Schapire et al., 1997). We will use the following identity

$$\mathcal{D}_{t+1}(i,k) = \frac{\exp\left(-\sum_{s=1}^{t}\alpha_s\rho(\widetilde{\mathsf{h}}_s^k,\mathbf{x}_i,\mathbf{y}_i)\right)}{ml\prod_{s=1}^{t}Z_s},$$

which can be straightforwardly derived from the expression of $\mathcal{D}_{t+1}$ in terms of $\mathcal{D}_t$. Then, we can write

$$\begin{aligned}
\widehat{R}_\rho\left(\frac{\widetilde{\mathsf{h}}}{\|\boldsymbol{\alpha}\|_1}\right) &= \frac{1}{ml}\sum_{i=1}^{m}\sum_{k=1}^{l}\mathbf{1}_{\rho(\widetilde{\mathsf{h}}^k,\mathbf{x}_i,\mathbf{y}_i)\leq\rho\|\boldsymbol{\alpha}\|_1}\\
&\leq \frac{1}{ml}\sum_{i=1}^{m}\sum_{k=1}^{l}\exp\left(-\rho(\widetilde{\mathsf{h}}^k,\mathbf{x}_i,\mathbf{y}_i)+\|\boldsymbol{\alpha}\|_1\rho\right)\\
&\leq \frac{1}{ml}e^{\|\boldsymbol{\alpha}\|_1\rho}\sum_{i=1}^{m}\sum_{k=1}^{l}\exp\Big(-\sum_{t=1}^{T}\alpha_t\rho(\widetilde{\mathsf{h}}_t^k,\mathbf{x}_i,\mathbf{y}_i)\Big)\\
&= e^{\|\boldsymbol{\alpha}\|_1\rho}\sum_{i=1}^{m}\sum_{k=1}^{l}\mathcal{D}_{T+1}(i,k)\prod_{t=1}^{T}Z_t\\
&= 2^T\prod_{t=1}^{T}\left[\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\right]^\rho\sqrt{\epsilon_t(1-\epsilon_t)},
\end{aligned}$$

where the first inequality holds by $1_{u\leq 0} \leq e^{-u}$ for all $u \in \mathbb{R}$ and the second by Jensen's inequality and the convexity of the maximum function. This concludes the proof of the theorem. □

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.

H. Additional experiments

In this Section we present additional experimental results that were not included in the main body of the paper due to space limitations.

H.1. Artificial data sets

The objective of the first artificial data set (ADS1) was to simulate the situation described in Section 1 where $h_1, \ldots, h_p$ are local experts. To generate the data we chose an arbitrary Markov chain over the English alphabet and sampled 40,000 random sequences each consisting of 10 symbols. For each sequence, we generated five expert predictions. Each expert was designed to have a certain probability of making a mistake at each position in the sequence. Expert h_j correctly predicted positions $2j-1$ and $2j$ with probability 0.97 and other positions with probability 0.5. We forced experts to make similar mistakes by making them select an adjacent alphabet symbol in case of an error. For example, when a mistake was made on a symbol b, the expert prediction was forced to be either a or c.

The second artificial data set (ADS2) modeled the case of rather poor experts. ADS2 was generated in the same way as ADS1, but expert predictions were different. This time each expert made mistakes at four of the ten distinct random positions in each sequence.

Table 4. Average Normalized Hamming Loss for ADS3. $\beta_{ADS1} = 0.95$, $\beta_{ADS2} = 0.95$, $T_{SLE} = 100$, $\delta = 0.05$.

$\mathcal{H}_{\text{MVote}}$	**0.1788 ± 0.00004**
$\mathcal{H}_{\text{FPL}}$	0.2189 ± 0.04097
$\mathcal{H}_{\text{CV}}$	**0.1788 ± 0.00004**
$\mathcal{H}_{\text{FPL-CV}}$	0.3148 ± 0.00387
$\mathcal{H}_{\text{ESPBoost}}$	0.1831 ± 0.00240
$\mathcal{H}_{\text{SLE}}$	0.1954 ± 0.00185
$\mathcal{H}_{\text{Rand}}$	0.3196 ± 0.00018
Best h_j	0.2957 ± 0.00005

The results on ADS1 and ADS2 can be found in Section 5. For all experiments with the algorithms $\mathcal{H}_{\text{Rand}}$, $\mathcal{H}_{\text{MVote}}$, and $\mathcal{H}_{\text{CV}}$ we ran the WMWP algorithm for $T = m$ rounds with the βs listed in the caption of Table 1 , generating distributions $\mathcal{P} \subseteq \{\mathsf{p}_1, \ldots, \mathsf{p}_T\}$. For $\mathcal{P}$ we used the collection of all suffix sets $\{\mathsf{p}_t, \ldots, \mathsf{p}_T\}$ and $\delta = 0.05$. For the algorithms based on FPL, we used $\epsilon = 0.5/pl$. The same parameter choices were used for the subsequent experiments.

In addition to ADS1 and ADS2, we also synthesized a third set. We simulated the case where each expert specialized in predicting some subset of the labels. In particular, we generated 40,000 random sequences over the English alphabet in the same way as for ADS1 and ADS2. To generate expert predictions, we partitioned the alphabet into 5 disjoint subsets A_j. Expert j always correctly predicted the label in A_j and the probability of correctly predicting the label not in A_j was set to 0.7. To train the ensemble algorithms, we used a training set of size $m = 200$.

The results are presented in Table 4. $\mathcal{H}_{\text{MVote}}$, $\mathcal{H}_{\text{CV}}$ and $\mathcal{H}_{\text{ESPBoost}}$ achieve the best performance on this data set with a considerable improvement in accuracy over the best expert h_j. We also observe as for the ADS2 experiment that $\mathcal{H}_{\text{Rand}}$ and $\mathcal{H}_{\text{FPL-CV}}$ fail to outperform the best model and approach the accuracy of the best path expert only asymptotically.

H.2. Pronunciation data sets

As pointed out in Section 5, it can be argued that for this task the edit-distance is a more suitable measure of performance than the average Hamming loss. Table 5 shows the results of our experiments. For these experiments, our ensemble algorithms were trained using the Hamming loss, but the performance is reported in terms of the edit-distance. For the SLE algorithm of Nguyen & Guo (2007) $\mathcal{H}_{\text{SLE}}$, the edit-distance was used for both training and testing. Remarkably, the results for edit-distance are comparable and $\mathcal{H}_{\text{MVote}}$ again offers the best performance despite not being optimized for this loss.

Finally, we also leveraged the fact that PDS2 is a larger data set to experiment with other training sizes. For the sake of completeness, the results are summarized in Table 6.

Table 5. Average edit distance, PDS1 and PDS2. $\beta_{PDS1} = 0.85$, $\beta_{PDS2} = 0.97$, $T_{SLE} = 100$, $\delta = 0.05$.

	PDS1, $m = 130$	PDS2, $m = 400$
$\mathcal{H}_{\text{MVote}}$	**0.8395 ± 0.01076**	**0.9626 ± 0.00341**
$\mathcal{H}_{\text{FPL}}$	1.0158 ± 0.34379	0.9744 ± 0.01277
$\mathcal{H}_{\text{CV}}$	**0.8668 ± 0.00553**	**0.9840 ± 0.00364**
$\mathcal{H}_{\text{FPL-CV}}$	1.8044 ± 0.09315	1.8625 ± 0.06016
$\mathcal{H}_{\text{ESPBoost}}$	1.3977 ± 0.06017	1.4092 ± 0.04352
$\mathcal{H}_{\text{SLE}}$	1.1762 ± 0.12530	1.2477 ± 0.12267
$\mathcal{H}_{\text{Rand}}$	1.8962 ± 0.01064	2.0838 ± 0.00518
Best h_j	1.2163 ± 0.00619	1.2883 ± 0.00219

Table 6. Average Hamming loss for PDS2. $\beta_{PDS2} = 0.97$, $T_{SLE} = 100$, $\delta = 0.05$

	$m = 200$	$m = 600$
$\mathcal{H}_{\text{MVote}}$	**0.2343 ± 0.00083**	**0.2304 ± 0.00148**
$\mathcal{H}_{\text{FPL}}$	**0.2393 ± 0.00335**	**0.2332 ± 0.00313**
$\mathcal{H}_{\text{CV}}$	**0.2364 ± 0.00048**	**0.2362 ± 0.00109**
$\mathcal{H}_{\text{FPL-CV}}$	0.4464 ± 0.01110	0.4063 ± 0.00976
$\mathcal{H}_{\text{ESPBoost}}$	0.3524 ± 0.00662	0.3458 ± 0.00276
$\mathcal{H}_{\text{SLE}}$	0.3217 ± 0.03929	0.3307 ± 0.03165
$\mathcal{H}_{\text{Rand}}$	0.4651 ± 0.00092	0.4544 ± 0.00308
Best h_j	0.3413 ± 0.00050	0.3412 ± 0.00053

H.3. OCR data set

Table 7 summarizes our results with the OCR data set. As can be seen from the table, the performance improvements of ensemble methods over the single best hypothesis are not statistically significant here.

H.4. Penn Treebank data set

To speed up the testing phase we only used sentences with less than 20 words (a total of 87,704 sentences).

For the second experiment (TR2) we trained 5 SVM^{struct} models. The five training sets (8,000 sentences each) were carefully chosen so that each contained the 8 most frequent POS tags but omitted a subset of some less frequent ones.

For the SVM algorithms, we generated 267,214 bag-of-word binary features. We first extracted all prefixes and suffices of length 2, 3, 4, 5 of all words in the data set. We then used binary features to indicate whether a given word contains one of the prefixes or suffices found. In addition, we also used features indicating whether preceding or following word contains one of those prefixes or suffices.

I. Example of sub-optimality of the SLE algorithm

In this section, we give an explicit construction showing that the SLE algorithm of Nguyen & Guo (2007) may produce ensembles that perform no better than the best expert

Table 7. Average Normalized Hamming Loss for OCR. $\beta = 0.5$, $T_{SLE} = 100$, $\delta = 0.05$.

$\mathcal{H}_{\text{MVote}}$	0.1992 ± 0.00274
$\mathcal{H}_{\text{FPL}}$	0.1992 ± 0.00270
$\mathcal{H}_{\text{CV}}$	0.1993 ± 0.00266
$\mathcal{H}_{\text{FPL-CV}}$	0.2030 ± 0.00278
$\mathcal{H}_{\text{ESPBoost}}$	0.1992 ± 0.00274
$\mathcal{H}_{\text{SLE}}$	0.1994 ± 0.00307
$\mathcal{H}_{\text{Rand}}$	0.1994 ± 0.00276
Best h_j	0.1994 ± 0.00306

h_j, which can be significantly worse than the performance of the optimal ensemble. We assume that $l = p = 2$, that $\mathcal{Y}$ is a space of binary sequences, and that expert h_j always correctly predicts the jth substructure. The probability of the event $\{h_1^2(\mathbf{x}) \neq y^2, h_2^1(\mathbf{x}) \neq y^1\}$ is set to be equal to q.

Suppose that the ensemble produced by SLE algorithm consists of T_j copies of expert h_j. If $T_1 < T_2$, then the SLE prediction always agrees with expert h_2. Conversely, if $T_1 > T_2$ then SLE prediction always agrees with expert h_1. Finally, if $T_1 = T_2$ then with probability p, the predictions of h_1 and h_2 disagree at both position 1 and 2 and, by definition of the algorithm, exactly one of these predictions must be chosen. In each of the cases above, the expected loss of the algorithm is bounded below by $q/2$. Since in our construction h_1 and h_2 can be chosen to have expected loss precisely $q/2$, we conclude that for this example the SLE algorithm produces ensembles that perform no better than the best expert h_j.

Note that in the above we can select $q = 1$, which, will result in an the expected loss of the SLE algorithm being 0.5, while an optimal ensemble for this problem can achieve 100% accuracy.

J. Discussion of other related work

In this section, we briefly discuss several other publications somewhat related to the topic of our work.

In the learning scenario we consider, 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, which are effectively viewed as black boxes. This scenario covers both the case where $h_1, \ldots, h_p$ are hand-crafted prediction rules and the one where they are the hypotheses returned by some learning algorithms trained on samples typically no longer available.

In contrast, most ensemble methods for structured prediction previously presented in the machine learning literature focus on scenarios where the learner can exploit some specific structure of the given experts $h_1, \ldots, h_p$ or where these experts are trained at the same time as the ensemble learner itself (Grubb et al., 2013; Payet & Todorovic, 2010; Tu & Bai, 2010). For example, the weak predictors used in the StructuredSpeedBoost algorithm of Grubb et al. (2013) have a very specific structure based on special selection and update rules. Similarly, the RF^2 algorithm of Payet & Todorovic (2010) uses tree experts to make its predictions. Finally, the auto-context algorithm of Tu & Bai (2010) is based on experts that are assumed to be probabilistic models.

Appendix References

Grubb, A., Munoz, D., Bagnell, J.A., and Hebert, M. Speedmachines: Anytime structured prediction. In *Budgeted Learning Workshop, ICML 2013*, 2013.

Payet, N. and Todorovic, S. (RF)^2 - Random Forest Random Field. In *Advances in NIPS*, pp. 1885–1893, 2010.

Tu, Z. and Bai, X. Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (10):1744–1757, Oct 2010.