Predicting Switching Graph Labelings with Cluster Specialists

Mark Herbster M.HERBSTER@CS.UCL.AC.UK and James Robinson J.ROBINSON@CS.UCL.AC.UK

Department of Computer Science
University College London
London
United Kingdom

Abstract

We address the problem of predicting the labeling of a graph in an online setting when the labeling is changing over time. Our primary algorithm is based on a specialist (Freund et al., 1997) approach; we develop the machinery of cluster specialists which probabilistically exploits the cluster structure in the graph. We show that one variant of this algorithm surprisingly only requires $O(\log n)$ time on any trial $t$ on an $n$-vertex graph. Our secondary algorithm is a quasi-Bayesian classifier which requires $O(t \log n)$ time to predict at trial $t$. We prove switching mistake-bound guarantees for both algorithms. For our primary algorithm, the switching guarantee smoothly varies with the magnitude of the change between successive labelings. In preliminary experiments we compare the performance of these algorithms against an existing algorithm (a kernelized Perceptron) and show that our algorithms perform better on synthetic data.

Keywords: online learning, experts, specialists, graph prediction, switching

1. Introduction

We study the problem of predicting a switching graph labeling. Consider the following game for predicting the labeling of a graph in the online setting. Nature presents a graph $G$; Nature queries a vertex $i_1 \in V = \{1, 2, \ldots, n\}$; the learner predicts the label of the vertex $\hat{y}_1 \in \{-1, 1\}$; Nature presents a label $y_1$; Nature queries a vertex $i_2$; the learner predicts $\hat{y}_2$; and so forth. The learner’s goal is to minimize the total number of mistakes $M = |\{t : \hat{y}_t \neq y_t\}|$. If Nature is strictly adversarial, the learner will incur a mistake on every trial, but if Nature is regular or simple, there is hope that the learner may incur only a few mistakes. Thus, a central goal of mistake-bounded online learning is to design algorithms whose total mistakes can be bounded relative to the complexity of nature’s labeling. This (non-switching) graph labeling problem has been studied extensively in the online learning literature (Herbster et al., 2008; Herbster and Lever, 2009; Cesa-Bianchi et al., 2013; Vitale et al., 2011; Herbster et al., 2015a). In this paper we generalize the setting to allow the underlying labeling to change arbitrarily over time. The learner has no knowledge of when a change in labeling will occur and therefore must be able to adapt quickly to these changes. Consider an example of a graph of major road junctions (vertices) connected by roads (edges), in which we want to predict whether or not a vertex is congested at any given time. Traffic congestion is naturally non-stationary with both gradual and abrupt changes occurring to the structure of the labeling over time (Kerner, 1998).

The structure of this paper is as follows. In Section 2 we give notation as well as discuss the background literature. In Section 3 we present a quasi-Bayes classifier as a baseline solution to the
switching graph-labeling problem. This algorithm has a stronger worst-case mistake bound under arbitrary changes to the labeling over time than our primary algorithm, however its per-trial time complexity is significantly larger, scaling linearly with $T$. In Section 4 we present the SWITCHING CLUSTER SPECIALISTS algorithm (SCS), a modification of the method of specialists (Freund et al., 1997) with the novel machinery of cluster specialists, a set of specialists that in a rough sense correspond to clusters in the graph. We consider two distinct sets of specialists, $B_n$ and $F_n$, where $B_n \subset F_n$. With the smaller set of specialists the bound is only larger by factor of $\log n$. On the other hand, prediction is exponentially faster per trial, remarkably requiring only $O(\log n)$ time to predict. The mistake bound analysis is more refined compared to the analysis of the quasi-Bayes classifier, in that the cost per switch models the magnitude of the change in structure of the labeling in the bound. In Section 5 we provide experiments on simulated non-stationary data. In Section 6 we provide some concluding remarks. All proofs are contained in the technical appendices.

2. Background

We first present common notation and then we review the background literature. Let $G = (V, E)$ be an undirected, connected, $n$-vertex graph with vertex set $V = \{1, 2, \ldots , n\}$ and edge set $E$. Each vertex of this graph may be labeled with one of two states $\{-1, 1\}$ and thus a labeling of a graph may be denoted by a vector $u \in \{-1, 1\}^n$ where $u_i$ denotes the label of vertex $i$. The cut-size of a labeling $u$ is defined as $\Phi_G(u) := |\{(i, j) \in E : u_i \neq u_j\}|$, i.e., the number of edges between vertices of disagreeing labels. We let $r_G(i, j)$ denote the resistance distance (effective resistance) between vertices $i$ and $j$ when the graph $G$ is seen as a circuit where each edge has unit resistance (e.g., Klein and Randić (1993)). The resistance diameter of a graph is $R_G := \max_{i,j \in V} r_G(i, j)$. The resistance weighted cut-size of labeling $u$ is $\Phi'_G(u) := \sum_{(i,j) \in E : u_i \neq u_j} r_G(i, j)$. Let $\Delta_n = \{\mu \in [0, 1]^n : \sum_{i=1}^n \mu_i = 1\}$ be the $n$-dimensional probability simplex. For $\mu \in \Delta_n$ we define $H(\mu) := \sum_{i=1}^n \mu_i \log_2 \frac{\mu_i}{\mu}$ to be the entropy of $\mu$. For $\mu, \omega \in \Delta_n$ we define $d(\mu, \omega) = \sum_{i=1}^n \mu_i \log_2 \frac{\mu_i}{\omega_i}$ to be the relative entropy between $\mu$ and $\omega$. For a vector $\omega$ and a set of indices $I$ let $\omega(I) := \sum_{i \in I} \omega_i$. For any positive integer $N$ we define $[N] := \{1, 2, \ldots , N\}$ and for any predicate $[\text{PRED}] := 1$ if $\text{PRED}$ is true and equals 0 otherwise.

2.1. Predicting the labeling of a graph

The problem of predicting the labeling of a graph in the batch setting was introduced as a foundational method for semi-supervised (transductive) learning. In this work, the graph was built using both the unlabeled and labeled instances. The seminal work by (Blum and Chawla, 2001) used a metric on the instance space and then built a kNN or $\epsilon$-ball graph. The partial labeling was then extended to the complete graph by solving a mincut-maxflow problem where opposing binary labels represented sources and sinks. In practice this method suffered from very unbalanced cuts. Significant practical and theoretical advances were made by replacing the mincut/maxflow model with methods based on minimising a quadratic form of the graph Laplacian. Influential early results include but are not limited to (Zhu et al., 2003; Belkin and Niyogi, 2004; Zhou et al., 2003). A limitation of the graph Laplacian-based techniques is that these batch methods—depending on their implementation—typically require $\Theta(n^2)$ to $\Theta(n^3)$ time to produce a single set of predictions. In the
Predicting the labeling of a graph in an online setting was introduced by Herbster et al. (2005). The authors proved bounds for a Perceptron-like algorithm with a kernel based on the graph Laplacian. Since this work there has been a number of extensions and improvements in bounds including but not limited to (Herbster et al., 2008; Cesa-Bianchi et al., 2009; Herbster and Lever, 2009; Herbster et al., 2015b,a; Rakhlin and Sridharan, 2017). Common to all of these papers is that a dominant term in their mistake bounds is the (resistance-weighted) cut-size.

From a simplified perspective, the methods for predicting the labeling of a graph (online) split into two approaches. The first approach works directly with the original graph and is usually based on a graph Laplacian (Herbster et al., 2005; Herbster and Lever, 2009; Herbster et al., 2015a); it provides bounds that utilize the additional connectivity of non-tree graphs, which are particularly strong when the graph contains uniformly-labeled clusters of small (resistance) diameter. The drawbacks of this approach are that the bounds are weaker on graphs with large diameter, and that computation times are slower.

The second approach is to approximate the original graph with an appropriately selected tree or “line” graph (Herbster et al., 2008; Cesa-Bianchi et al., 2013, 2009; Vitale et al., 2011). This enables faster computation times, and bounds that are better on graphs with large diameters. These algorithms may be extended to non-tree graphs by first selecting a spanning tree uniformly at random (Cesa-Bianchi et al., 2013) and then applying the algorithm to the sampled tree. This randomized approach induces expected mistake bounds that also exploit the cluster structure in the graph (see Section 2.3). Our algorithms take this approach.

### 2.2. Switching Prediction

In this paper rather than predicting a single labeling of a graph we instead will predict a (switching) sequence of labelings. Switching in the mistake- or regret-bound setting refers to the problem of predicting an online sequence when the “best comparator” is changing over time. In the simplest of switching models the set of comparators is structureless and we simply pay per switch. A prominent early result in this model was given by Herbster and Warmuth (1998). This paper introduced the fixed-share update which will play a prominent role in our main algorithm. Other prominent results in the structureless model include but are not limited to (Vovk, 1999; Bousquet and Warmuth, 2002; György et al., 2005; Koolen and Rooij, 2008; Koolen et al., 2012; Cesa-Bianchi et al., 2012). A stronger model is to instead prove a bound that holds for any arbitrary contiguous sequence of trials. Such a bound is called an adaptive-regret bound. This type of bound automatically implies a bound on the structureless switching model. Adaptive-regret was introduced in (Hazan and Seshadhri, 2007)1 other prominent results in this model include (Adamskiy et al., 2012; Cesa-Bianchi et al., 2012; Daniely et al., 2015).

The structureless model may be generalized, by introducing a divergence measure on the set of comparators. Thus, whereas in the structureless model we pay for the number of switches, in the structured model we instead pay in the sum of divergences between successive comparators. This model was introduced in (Herbster and Warmuth, 2001); prominent results include (Kivinen et al., 2004; Cesa-Bianchi et al., 2012).

---

1. However, see the analysis of WML in (Littlestone and Warmuth, 1994) for a precursory result.
The results most directly comparable to ours are contained in (Herbster et al., 2015b). That paper also considers switching graph label prediction. However, the main results of that paper are not directly comparable to ours as they apply to the more combinatorially challenging problem of repeated switching within a small set of graph labelings contained in a larger set. That set-up was originally framed in the “experts” setting and posed as an open problem by Freund (2000) and solved in (Bousquet and Warmuth, 2002). If we consider the bound in (Herbster et al., 2015b) applied to the case where there is not repeated switching within a smaller set, then their bound is uniformly and significantly weaker than the bounds in this paper and the algorithm is quite slow requiring \( \theta(n^3) \) time per trial in a typical implementation. Also contained in (Herbster et al., 2015b) is a baseline algorithm based on a kernel perceptron with a graph Laplacian kernel. The bound of that algorithm has the significant drawback in that it scales with respect to the “worst” labeling in a sequence of labelings. However, it is simple to implement and we use it as a benchmark in our experiments.

2.3. Random Spanning Trees and Linearization

Since we operate in the transductive setting where the entire unlabeled graph is presented to the learner beforehand, this affords the learner the ability to perform any reconfiguration to the graph as a preprocessing step. The bounds of most existing algorithms for predicting a labeling on a graph are usually expressed in terms of the cut-size of the graph under that labeling. A natural approach then is to use a spanning tree of the original graph which can only reduce the cut-size of the labeling.

The effective resistance between vertices \( i \) and \( j \), denoted \( r_G(i, j) \), is equal to the probability that a spanning tree of \( G \) drawn uniformly at random (from the set of all spanning trees of \( G \)) includes \((i, j) \in E\) as one of its \( n - 1 \) edges (e.g., Lyons and Peres (2017)). As first observed by Cesa-Bianchi et al. (2009), by selecting a spanning tree uniformly at random from the set of all possible spanning trees, mistake-bounds expressed in terms of the cut-size then become expected mistake bounds now in terms of the effective-resistance-weighted cut-size of the graph. That is, if \( R \) is a random spanning tree of \( G \) then \( \mathbb{E}[\Phi_R(u)] = \Phi^r_G(u) \) and thus \( \Phi^r_G(u) \leq \Phi_G(u) \).

To illustrate the power of this randomization consider the simplified example of a graph with two cliques each of size \( n/2 \), where one clique is labeled uniformly with ‘+1’ and the other ‘-1’ with an additional arbitrary \( n/2 \) “cut” edges between the cliques. This dense graph exhibits two disjoint clusters and \( \Phi_G(u) = n/2 \). On the other hand \( \Phi^r_G(u) = \Theta(1) \), since between any two vertices in the opposing cliques there are \( \frac{n}{2} \) edge disjoint paths of length \( \leq 3 \) and thus the effective resistance between any pair of vertices is \( \Theta(1/n) \). Since bounds usually scale linearly with (resistance-weighted) cut-size, the cut-size bound would be vacuous but the resistance-weighted cut-size bound would be small.

Both of our algorithms (see Sections 3 and 4) make use of this preprocessing step of sampling a uniform random spanning tree, as well as a linearization of this tree to produce a (spine) line-graph, \( S \). A random spanning tree can be sampled from a graph efficiently using a random walk or similar methods (see e.g., Wilson (1996)). The linearization of \( G \) to \( S \) as a preprocessing step was first proposed by Herbster et al. (2008) and has since been applied in, e.g., (Cesa-Bianchi et al., 2013; Padilla et al., 2018). In order to construct \( S \), a random-spanning tree \( R \) is picked uniformly at random. A vertex of \( R \) is then chosen and the graph is fully traversed using a depth-first search generating an ordered list \( V_L = \{i_1, \ldots, i_{2m+1}\} \) of vertices in the order they were visited. Vertices
in \( V \) may appear multiple times in \( V_L \). A subsequence \( V_L' \subseteq V_L \) is then chosen such that each vertex in \( V \) appears only once. The line graph \( S \) is then formed by connecting each vertex in \( V_L' \) to its immediate neighbors in \( V_L' \) with an edge. We denote the edge set of \( S \) by \( E_S \). Surprisingly, as stated in the lemma below, the cut on this linearized graph is no more than twice the cut on the original graph.

**Lemma 1 (Herbster et al., 2008)** Given a labeling \( u \in \{-1, 1\}^n \) on a graph \( G \), for the mapping \( G \to R \to S \), as above, we have \( \Phi_S(u) \leq 2\Phi_R(u) \leq 2\Phi_G(u) \).

By combining the above observations we may reduce the problem of learning on a graph to that of learning on a line graph. In particular, if we have an algorithm with a mistake bound of the form \( M \leq O(\Phi_G(u)) \) this implies we then may give an expected mistake bound of the form \( M \leq O(\Phi_S(u)) \) by first sampling a random spanning tree and then linearizing it as above. Thus, for simplicity in presentation, we will only state the deterministic mistake bounds in terms of cut-size, although the expected bounds in terms of resistance-weighted cut-sizes will hold simultaneously.

Before presenting our algorithms we first introduce the following useful notation. The underlying assumption is that we are predicting vertex labels from a sequence \( u_1, \ldots, u_T \in \{-1, 1\}^n \) of graph labelings over \( T \) trials. The set \( K := \{t \in \{2, \ldots, T\} : u_t \neq u_{t-1} \} \cup \{1\} \) contains the first trial of each of the \( |K| \) “segments” of the prediction problem. Each segment corresponds to a time period when the underlying labeling is unchanging. We let \( \Phi_t := \Phi(u_t) \), where the cut \( \Phi \) is with respect to the linear embedding \( S \). Finally, we define the mean cut of the distinct labelings as \( \bar{\Phi} := \sum_{k \in K} \Phi_k / |K| \).

### 3. A Quasi-Bayes Classifier

We first present a Bayesian classifier under the following model. The Ising distribution \( p_G(u) \propto e^{-\Phi_G(u)/T} \) assigns to a graph labeling \( u \) a probability that is exponentially decreasing with respect to its cut-size. As such it is natural to use this distribution for graph label prediction with an inductive bias towards small cut-size labelings. However, if we wish to compute marginal probabilities to predict labels then exact computation is known to be NP-hard for general graphs (Provan and Ball, 1983). This motivates our approach to randomly select a spine \( S \subseteq G \). Given this linearization we may re-parameterize \( T \) in terms of \( \theta \) so that

\[
p_S(u) := \frac{1}{2} \theta^{\phi_S(u)} (1 - \theta)^{n-1 - \phi_S(u)},
\]

where \( \theta \) is now the probability that an edge is cut between any pair of adjacent vertices on the spine. In this linearized model we can compute the marginal probability of a label at a vertex given the observed label sequence in \( O(\log n) \) time in an online setting (Herbster et al., 2008).

We may then extend (1) to a probability distribution \( p_S^\Sigma \) over sequences of graph labelings \( u_1, \ldots, u_T \) as follows. The initial graph labeling \( u_1 \) is drawn from (1); the remaining labelings are generated by introducing a sequence of latent switch-reset (Bracegirdle and Barber, 2011) Bernoulli random variables \( \langle S_1, \ldots, S_{T-1} \rangle \). A switch-reset occurs at the end of trial \( t \) if \( S_t = 1 \) with \( \mathbb{P}(S_t = 1) := \alpha \). When a switch-reset occurs a (potentially) new labeling \( u_{t+1} \) is drawn from (1). Thus \( u_{t+1} = u_t \) with probability \( (1 - \alpha) + \alpha p_S(u_t) \), otherwise a labeling \( u_{t+1} \neq u_t \) is generated with probability \( \alpha (1 - p_S(u_t)) \).
The online prediction on trial $t$ of vertex $i_t$, given the sequence $(i_1, y_1), \ldots, (i_{t-1}, y_{t-1})$ of observed vertex-label pairs, is

$$\hat{y}_t^{\text{sa}} := \arg\max_{y \in \{-1,1\}} p_{S}^{\text{sa}}(u_{i_t} = y | u_{i_1} = y_1, \ldots, u_{i_{t-1}} = y_{t-1}).$$

Thus, to predict the label of $i_t$ we simply predict the label with maximum marginal probability. By successively predicting the label of maximum marginal probability we can bound the number of mistakes of this algorithm by $M^{\text{sa}} \leq -\log_2 (p_{S}^{\text{sa}}(u_{i_1} = y_1, \ldots, u_{i_T} = y_T))$ via the standard Halving algorithm analysis (Barzdin and Freivald, 1972). We may lower bound the probability of a given sequence of labelings with $|K| - 1$ switches, $p_{S}^{\text{sa}}(u_1, \ldots, u_T)$ with

$$\alpha^{|K|-1} (1 - \alpha)^{T-|K|} \prod_{k \in K} p(u_k) = \alpha^{|K|-1} (1 - \alpha)^{T-|K|} 2^{-|K|} \Phi(1 - \theta)^{K(n-1-\Phi)}.$$  \hspace{0.5cm} (2)

The classifier $\hat{y}_t^{\text{sa}}$ minimizes the Bayes risk with respect to the 0–1 loss. However, in the following theorem we will exploit the fact that by making the classifier conservative we may reduce the $\log T$ term appearing in the mistake bound to $\log \log T$ (see Appendix A for details). A conservative classifier only modifies its internal state on the trials in which a mistake is incurred. Thus in this case when we marginalize we only do so with respect to the past examples which were mispredicted and this is why we say the method is “quasi-Bayesian”. Now if we predict with CONSERVATIVE-$\hat{y}_t^{\text{sa}}$ we have from the point of view of the algorithm’s state that $T = M_B$, where $M_B$ denotes the number of mistakes incurred by the algorithm. Then by selecting an optimal post-hoc tuning of $\theta$ and $\alpha$ and substituting into (2) we have the following theorem,

**Theorem 1** If the distribution $p_{S}^{\text{sa}}$ is parameterized with $\alpha := \frac{|K|-1}{M_B - 1}$ and $\theta := \Phi/(n - 1)$ then the number of mistakes, $M_B$, made in predicting the online sequence $(i_1, y_1), \ldots, (i_T, y_T)$ with CONSERVATIVE-$\hat{y}_t^{\text{sa}}$ (Q-BAY) is upper bounded by

$$O \left( \sum_{k \in K} \Phi_k \log(n/\Phi) + |K| \log |K| + |K| \log \log T \right),$$

for any sequence of labelings $u_1, \ldots, u_T \in \{-1,1\}^n$ such that $u_{t,i_t} = y_t$ for all $t \in [T]$.

See Appendix A for a proof and see Appendix B for details of the following proposition.

**Proposition 2** The quasi-Bayes classifier CONSERVATIVE-$\hat{y}_t^{\text{sa}}$ (Q-BAY) may be computed on trial $t$ in $O(t \log n)$ time, requiring $O(tn)$ space.

The quasi-Bayes classifier gives a good mistake-bound guarantee, however the leading term in the mistake bound scales with the sum of cut-sizes of each labeling. This means we potentially overpay on a smoothly-changing sequence of labelings, where the graph labeling changes gradually from one switch to the next. The algorithm also requires $O(t \log n)$ time on trial $t$. In the next section we improve on both of these issues with our main algorithm; one variant of which boasts a $O(\log n)$ per-trial time-complexity and a mistake bound which is a function of a summation of a Hamming-like divergence measure between consecutive labelings.
Algorithm 1: Switching Cluster Specialists

**input**: Specialists set \( \mathcal{E} \)

**parameter**: \( \alpha \in [0, 1] \)

**initialize**: \( \omega_1 \leftarrow \frac{1}{|\mathcal{E}|}, \omega_0 \leftarrow \frac{1}{|\mathcal{E}|}, p \leftarrow 0, m \leftarrow 0 \)

for \( t = 1 \) to \( T \) do
  
  receive \( i_t \in V \)
  
  set \( \mathcal{A}_t := \{ \varepsilon \in \mathcal{E} : \varepsilon(i_t) \neq \Box \} \)

  foreach \( \varepsilon \in \mathcal{A}_t \) do
    
    \[
    \omega_{t,\varepsilon} \leftarrow (1 - \alpha)^{m - p_{\varepsilon}} \omega_{t-1,\varepsilon} + \frac{1 - (1 - \alpha)^{m - p_{\varepsilon}}}{|\mathcal{E}|} \]  
    
  predict \( \hat{y}_t \leftarrow \text{sign} \left( \sum_{\varepsilon \in \mathcal{A}_t} \omega_{t,\varepsilon} \varepsilon(i_t) \right) \)

  receive \( y_t \in \{-1, 1\} \)

  set \( \mathcal{Y}_t := \{ \varepsilon \in \mathcal{E} : \varepsilon(i_t) = y_t \} \)

  if \( \hat{y}_t \neq y_t \) then
    
    \[
    \omega_{t,\varepsilon} \leftarrow \begin{cases} 
    0 & \varepsilon \in \mathcal{A}_t \cap \mathcal{Y}_t \\
    \omega_{t-1,\varepsilon} & \varepsilon \notin \mathcal{A}_t \\
    \frac{\omega_{t,\varepsilon}^{(i_t)}}{\omega_t^{(\mathcal{Y}_t)}} & \varepsilon \in \mathcal{Y}_t 
    \end{cases} \]  
    
    foreach \( \varepsilon \in \mathcal{A}_t \) do
      
      \[
      p_{\varepsilon} \leftarrow m \\
      m \leftarrow m + 1 
      \]
    
  else
    
    \( \omega_t \leftarrow \omega_{t-1} \)

4. Switching Specialists

In this section we present a new method based on the idea of specialists (Freund et al., 1997) from the prediction with expert advice literature (Littlestone and Warmuth, 1994; Vovk, 1990; Cesa-Bianchi and Lugosi, 2006). Although the achieved bounds are slightly worse than other methods for predicting a single labeling of a graph, the derived advantage is that it is possible to combine these techniques with a fixed-share update (Herbster and Warmuth, 1998) in order to obtain “competitive” bounds with fast algorithms to predict a sequence of graph labelings.

Our inductive bias is to predict well when a labeling has a small (resistance-weighted) cut-size. The complementary perspective implies that the labeling consists of a few uniformly labeled clusters. This suggests the idea of maintaining a collection of basis functions where each such function is specialized to predict a constant function on a given cluster of vertices. To accomplish this technically we adapt the method of specialists (Freund et al., 1997; Koolen et al., 2012). A specialist is a prediction function \( \varepsilon \) from an input space to an extended output space with abstentions. So for us the input space is just \( V = [n] \), the vertices of a graph; and the extended output space is \( \{-1, 1, \Box\} \) where \( \{-1, 1\} \) corresponds to predicted labels of the vertices, but ‘\( \Box \)’ indicates that the specialist abstains from predicting. Thus a specialist specializes its prediction to part of the input space and in our application the specialists correspond to a collection of clusters which cover the graph, each cluster uniformly predicting \(-1\) or \(1\).

In Algorithm 1 we give our switching specialists method. The algorithm maintains a weight vector \( \omega_t \) over the specialists in which the magnitudes may be interpreted as the current confidence.
we have in each of the specialists. The updates and their analyses are a combination of three standard methods: i) Halving loss updates, ii) specialists updates and iii) (delayed) fixed-share updates. The loss update (4) zeros the weight components of incorrectly predicting specialists, while the non-predicting specialists are not updated at all. In (3) we give our delayed fixed-share style update. A standard fixed share update may be written in the following form:

$$
\omega_{t,\varepsilon} = (1 - \alpha)\omega_{t-1,\varepsilon} + \frac{\alpha}{|E|}.
$$

Although (5) superficially appears different to (3), in fact these two updates are exactly the same in terms of predictions generated by the algorithm. This is because (3) caches updates until the given specialist is again active. The purpose of this computationally is that if the active specialists are, for example, logarithmic in size compared to the total specialist pool, we may then achieve an exponential speedup over (5); which in fact we will exploit.

In the following theorem we will give our switching specialist bound. The dominant cost of switching on trial $t$ to $t + 1$ is given by the non-symmetric $J_E(\mu_t, \mu_{t+1}) := |\{\varepsilon \in E : \mu_{t,\varepsilon} = 0, \mu_{t+1,\varepsilon} \neq 0\}|$, i.e., we pay only for each new specialist introduced but we do not pay for removing specialists.

**Theorem 3** For a given specialist set $E$, let $M_E$ denote the number of mistakes made in predicting the online sequence $(i_1, y_1), \ldots, (i_T, y_T)$ by Algorithm 1. Then,

$$
M_E \leq \frac{1}{\pi_1} \log |E| + \sum_{t=1}^{T} \frac{1}{\pi_t} \log \frac{1}{1 - \alpha} + \sum_{i=1}^{K-1} J_E(\mu_{k_i}, \mu_{k_{i+1}}) \log \frac{|E|}{\alpha},
$$

for any sequence of consistent and well-formed comparators $\mu_1, \ldots, \mu_T \in \Delta_{|E|}$ where $K := \{k_1 = 1 < \cdots < k_{|E|}\} := \{t \in [T]: \mu_t \neq \mu_{t-1}\} \cup \{1\}$, and $\pi_t := \mu_t(\mathcal{Y}_t)$.

The bound in the above theorem depends crucially on the best sequence of consistent and well-formed comparators $\mu_1, \ldots, \mu_T$. The consistency requirement implies that on every trial there is no active incorrect specialist assigned “mass” $(\mu_t(A_t \setminus \mathcal{Y}_t) = 0)$. We may eliminate the consistency requirement by “softening” the loss update (4). A comparator $\mu \in \Delta_{|E|}$ is well-formed if $\forall v \in V$, there exists a unique $\varepsilon \in E$ such that $\varepsilon(v) \neq \Box$ and $\mu_\varepsilon > 0$, and furthermore there exists a $\pi \in [0, 1]$ such that $\forall \varepsilon \in E : \mu_\varepsilon \in \{0, \pi\}$, i.e., each specialist in the support of $\mu$ has the same mass $\pi$ and these specialists disjointly cover the input space ($V$). At considerable complication to the form of the bound the well-formedness requirement may be eliminated.

The above bound is “smoother” in this theorem than in Theorem 1 as it now scales with a gradual change in the comparator. In the next section we describe the novel specialists sets that we’ve tailored to graph-label prediction so that a small change in comparator corresponds to a small change in a graph labeling.

### 4.1. Cluster Specialists

In order to construct the cluster specialists over a graph $G = (V = [n], E)$, we first construct a line graph as described in Section 2.3. A cluster specialist is then defined by $\varepsilon^L_y(\cdot)$ which maps $V \rightarrow \{-1, 1, \Box\}$ where $\varepsilon^L_y(v) := y$ if $l \leq v \leq r$ and $\varepsilon^L_y(v) := \Box$ otherwise. Hence cluster specialist $\varepsilon^L_y(v)$ just corresponds to a function that predicts the label $y$ if vertex $v$ lies between vertices $l$ and $r$ and abstains otherwise. Recall that by sampling a random spanning tree the expected cut-size of a labeling on the spine is no more than twice the resistance-weighted cut-size on $G$.  

8
Thus, given a labeled graph with a small resistance-weighted cut-size with densely interconnected clusters and modest intra-cluster connections, then by implication a cut-bracketed linear segment on the spine will in expectation roughly correspond to one of the original dense clusters. We will consider two basis sets of cluster specialists. The first, $\mathcal{F}_n$, contains all possible cluster specialists.

We define the complete basis set $\mathcal{F}_n := \{ \epsilon_{l,r}^{y} : l, r \in [n], l \leq r; y \in \{-1, 1\} \}$. We say that a set of specialists $C_u \subseteq \mathcal{E} \subseteq 2^{\{-1,1,\Box\}}$ from basis $\mathcal{E}$ covers a labeling $u \in \{-1, 1\}^n$ if for all $v \in V = [n]$ and $\epsilon \in C_u$ that $\epsilon(v) \in \{u_v, \Box\}$ and if $v \in V$ then there exists $\epsilon \in C_u$ such that $\epsilon(v) = u_v$. The basis $\mathcal{E}$ is complete if every labeling $u \in \{-1, 1\}^n$ is covered by some $C_u \subseteq \mathcal{E}$. The basis $\mathcal{F}_n$ is complete and in fact has the following approximation property: for any $u \in \{-1, 1\}^n$ there exists a covering set $C_u \subseteq \mathcal{F}_n$ such that $|C_u| = \Phi_S(u) + 1$. This follows directly as a line with $k - 1$ cuts is divided into $k$ segments. We now illustrate the use of basis $\mathcal{F}_n$ to predict the labeling of a graph. For simplicity we illustrate by considering the problem of predicting a single graph labeling without switching. As there is no switch we will set $\alpha := 0$ and thus if the graph is labeled with $u \in \{-1, 1\}^n$ with cut-size $\Phi_S(u)$ then we will need $\Phi_S(u) + 1$ specialists to predict the labeling and thus the comparators may be post-hoc optimally determined so that $\mu = \mu_1 = \cdots = \mu_{\Phi_S(u)}$ and there will be $\Phi_S(u) + 1$ components of $\mu$ each with “weight” $1/(\Phi_S(u) + 1)$, thus $1/\pi_1 = \Phi_S(u) + 1$, since there will be only one specialist (with non-zero weight) active per trial. Since the cardinality of $\mathcal{F}_n$ is $n^2 + n$, by substituting into (6) we have that the number of mistakes will be bounded by $(\Phi_S(u) + 1) \log(n^2 + n)$. Note for a single graph labeling on a spine this bound is not much worse than the best known result (Herbster et al., 2008, Theorem 4). In terms of computation time however it is significantly slower than the algorithm in (Herbster et al., 2008) requiring $\Theta(n^2)$ time to predict on a typical trial since on average there are $\Theta(n^2)$ specialists active per trial. We next introduce the basis $B_n$ which has $\Theta(n)$ specialists and only requires $O(\log n)$ time per trial to predict with only a small increase in bound.

The basis

$$B_{p,q} := \begin{cases} \{\epsilon_{-1}^{p,q}, \epsilon_{1}^{p,q}\} & p = q, \\ \{\epsilon_{-1}^{p,q}, \epsilon_{1}^{p,q}\} \cup B_{p,\lfloor \frac{p+q}{2} \rfloor} \cup B_{\lceil \frac{p+q}{2} \rceil, q} & p \neq q \end{cases}$$

is analogous to a binary tree. We have the following approximation property for $B_n := B_{1,1}$.

**Proposition 4** The basis $B_n$ is complete. Furthermore, for any labeling $u \in \{-1, 1\}^n$ there exists a covering set $C_u \subseteq B_n$ such that $|C_u| \leq 2(\Phi_S(u) + 1)\lceil \log_2 \frac{n}{9} \rceil$ for $n > 2$.

From a computational perspective the binary tree structure ensures that there are only $\Theta(\log n)$ specialists active per trial, leading to an exponential speed-up in prediction. A similar set of specialists were used for obtaining adaptive-regret bounds in (Daniely et al., 2015; Jun et al., 2017). In that work the “binary tree” structure is over the time dimension (trial sequence) whereas in this work the binary tree is over the space dimension (graph) and a fixed-share update is used to obtain adaptivity over the time dimension.\(^2\)

Similarly to Q-BAY in section 3, the SCS algorithm employs a conservative update, that is, the specialists’ weights are only updated on trials on which the algorithm makes a mistake. Again the motivation for this is that the conservative updating reduces the usual $\log T$ term induced by a fixed-share update to a $\log \log T$ term in the bounds given in Corollary 6.

\(^2\) An interesting open problem is to try to find good bounds and time-complexity with sets of specialists over both the time and space dimensions.
In the coming corollary we will give our bound for switching graph labelings. The bound is smooth as the cost per switch will be measured with a Hamming-like divergence \( H \) on the “cut” edges between successive labelings, defined as

\[
H(u, u') := \sum_{(i,j) \in E_S} \left[ \left[ u_i \neq u_j \right] \lor \left[ u'_i \neq u'_j \right] \right] \land \left[ \left[ u'_i \neq u'_j \right] \lor \left[ u_j \neq u'_j \right] \right].
\]

Observe that \( H(u, u') \) is no larger than twice the hamming distance between \( u \) and \( u' \) and is often significantly smaller. To achieve the coming bound we will need the following proposition, which upper bounds divergence \( J \) by \( H \), a subtlety is that there are many distinct sets of specialists consistent with a given comparator. For example, consider a uniformly labeled graph we may “cover” this graph with a single specialist or alternatively we may cover with a specialist for each vertex. For the sake of simplicity in bounds we will always choose the smallest set of covering specialists. Thus we introduce the notion of minimal-consistency. A comparator \( \mu \in \Delta_{|E|} \) is consistent with the labeling \( u \in \{-1, 1\}^n \) if \( \mu \) is well-formed and \( \mu_v > 0 \) implies for all \( v \in V \) that \( \epsilon(v) \in \{u_v, \square\} \); furthermore a comparator \( \mu \) is minimal-consistent with \( u \) if the cardinality of its support set \( \{\mu_v : \mu_v > 0\} \) is the minimum of all comparators consistent with \( u \).

**Proposition 5** For a linearized graph \( S \), for comparators \( \mu, \mu' \in \Delta_{|F_n|} \) that are minimal-consistent with \( u \) and \( u' \) respectively,

\[
J_{F_n}(\mu, \mu') \leq \min \left( 2H(u, u') \cdot \Phi_S(u') + 1 \right).
\]

A proof is given in Appendix E. In the following corollary we summarize the results of the SCS algorithm using the basis sets \( F_n \) and \( B_n \) with an optimally-tuned switching parameter \( \alpha \).

**Corollary 6** For a connected \( n \)-vertex graph \( G \) and with randomly sampled spine \( S \), the number of mistakes made in predicting the online sequence \((i_1, y_1), \ldots, (i_T, y_T)\) by the SCS algorithm with optimally-tuned \( \alpha \) is upper bounded with basis \( F_n \) by

\[
O \left( \Phi_1 \log n + \sum_{i=1}^{\lfloor K \rfloor - 1} H(u_{k_i}, u_{k_{i+1}}) \left( \log n + \log |K| + \log \log T \right) \right)
\]

and with basis \( B_n \) by

\[
O \left( \Phi_1 \log n + \sum_{i=1}^{\lfloor K \rfloor - 1} H(u_{k_i}, u_{k_{i+1}}) \left( \log n + \log |K| + \log \log T \right) \right) \log n
\]

for any sequence of labelings \( u_1, \ldots, u_T \in \{-1, 1\}^n \) such that \( u_{t,i_t} = y_t \) for all \( t \in [T] \).

Thus the bounds are equivalent up to a factor of \( \log n \) although the computation times vary dramatically. See Appendix F for a technical proof of these results, and details on the selection of the switching parameter \( \alpha \). Note that we may avoid the issue of needing to optimally tune \( \alpha \) using the following method proposed by Herbster (1997) and by Koolen and Rooij (2008). We use a time-varying parameter and on trial \( t \) we set \( \alpha_t = \frac{1}{t + 1} \). We have the following guarantee for this method, see Appendix G for a proof.

**Proposition 7** For a connected \( n \)-vertex graph \( G \) and with randomly sampled spine \( S \), the SCS algorithm with bases \( F_n \) and \( B_n \) when predicting the online sequence \((i_1, y_1), \ldots, (i_T, y_T)\) now with time-varying \( \alpha \) set equal to \( \frac{1}{t + 1} \) on trial \( t \) achieves the same asymptotic mistake bounds as in Corollary 6 with an optimally-tuned \( \alpha \), under the assumption that \( \Phi_S(u_1) \leq \sum_{i=1}^{\lfloor K \rfloor - 1} J_S(\mu_{k_i}, \mu_{k_{i+1}}) \).
an additional significant term to the mistake bound of SCS-B. The following simple proposition
produces the mistake bound of Corollary 6 with an additional

\[ r \mathbb{P}(\epsilon) \leq \frac{1}{2} \sum_{i=1}^{r} M_i \]

and Schapire, 1997) since the latter would introduce another parameter to tune, as well as
weighted majority vote was chosen over running a meta-algorithm on top such as e.g., Hedge (Fre-
and linearized following the procedure in Section 2.3. Aggregating ensemble predictions by an un-
odd numbers to avoid ties. For each instance a spanning tree was first drawn uniformly at random
of mistakes made by aggregating predictions by an unweighted majority vote on each trial
\[ t \]

For algorithms SCS-F, SCS-B, and Q-BAY we tested ensemble sizes in

\[ \{3, 5, 9, 17, 33\} \]

performance of using ensembles of instances of our algorithms, aggregated in the same fashion.
Ensemble size

\[ k \]-nearest neighbors graph (we used

\[ k = 4096 \]

To emulate a non-stationary setting we assigned labels in the following fashion. To generate
a binary labeling we sampled without replacement 5 digits from \( \{0, \ldots, 9\} \), assigning these to
class ‘+1’ and the remaining 5 digits to class ‘-1’. For each experiment the number of trials was
\( T = 1000 \), on each trial a vertex from the graph was sampled, then the four algorithms provided
predictions of the current label of the vertex. Every 100 trials we generated a new binary labeling
by the above procedure, thus each experiment consisted of 10 random labelings.

Following the experiments of Cesa-Bianchi et al. (2013) in which ensembles of random span-
ning trees were drawn and aggregated by an unweighted majority vote, we tested the effect in
performance of using ensembles of instances of our algorithms, aggregated in the same fashion. For algorithms SCS-F, SCS-B, and Q-BAY we tested ensemble sizes in \( \{1, 3, 5, 9, 17, 33\} \), using odd numbers to avoid ties. For each instance a spanning tree was first drawn uniformly at random
and linearized following the procedure in Section 2.3. Aggregating ensemble predictions by an un-
weighted majority vote was chosen over running a meta-algorithm on top such as e.g., Hedge (Fre-
und and Schapire, 1997) since the latter would introduce another parameter to tune, as well as
produce the mistake bound of Corollary 6 with an additional \( O(\log r) \) term for an ensemble of size
\( r \). Alternatively for the SCS algorithms the \( r \) independent specialist sets could have been combined
into one large pool and Algorithm 1 used on this pool. This method would however also introduce
an additional significant term to the mistake bound of SCS-B. The following simple proposition
bounds the total number of mistakes made by taking the unweighted majority vote, a proof is given
in Appendix I.

**Proposition 8** For an ensemble of \( r \) algorithms with mistake bounds \( M_1, \ldots, M_r \), the number
of mistakes made by aggregating predictions by an unweighted majority vote on each trial \( t = 1, \ldots, T \) is bounded by

\[ 2(\sum_{i=1}^{r} M_i)/r. \]
Figure 1: Mean cumulative error over 12 iterations of 10 switches every 100 trials on an $n = 4096$ graph, with $k = 3$ and ensemble sizes in \{1, 33\}.

Each experiment was repeated over 10 iterations and the mean cumulative error of each algorithm for the largest and smallest ensemble sizes is plotted in Figure 1 for $k = 3$; then in Table 1 we give the final mean errors for $k = 3, 12$ and the intermediate ensemble sizes.

The parameter $\alpha$ was set to $\sum_{k\in\mathcal{K}}\{1\} (\Phi_k + 1)/\sum_{t=2}^T (\Phi_t + 1)$ as an approximation to the optimal value for both SCS-F and SCS-B (cut sizes are measured on the spine, $S$). The parameters $\alpha$ and $\theta$ for Q-BAY were set to $(|\mathcal{K}| - 1)/(T - 1)$, and $\bar{\Phi}/|E|$, where $|E|$ is the number of edges in $G$. The parameter $\gamma$ in SGP was set to $\max_{k\in\mathcal{K}} \|u_k\|_K$ (see Appendix H). There are several observations to be made from our experiments. Firstly, the use of ensembles showed a drastic improvement in the performance of the three algorithms which used a linear graph embedding, as shown in Table 1. The performance of the SCS-B algorithm was also competitive compared to that of SCS-F, despite having the weaker guarantee, also shown in Figure 1. The effect of increasing $k$ when producing the $k$-nearest neighbors graph hurt the performance of all four methods, but the effect was much stronger on the linearized algorithms, and less so on the SGP.

6. Conclusion

Our primary result was an algorithm for switching graph labelings with a per-trial prediction time of $O(\log n)$ and a mistake bound that smoothly tracks changes to the graph labeling over time. In the long version of this paper we plan to extend the analysis of the primary algorithm to the expected regret setting; we anticipate no technical difficulties just a certain long-form non-tidiness to the bound due to relaxing our simplifying assumption of the well-formed comparator sequence that is minimal-consistent with the labeling sequence. From a technical perspective two of the open problems that we found intriguing was first to extend the analysis of Q-BAY to the smooth setting in terms of $H(\cdot, \cdot)$ and second to eliminate the $\log \log T$ term from our bounds. Regarding the second problem, the natural approach would be to replace the conservative fixed-share update with a variable-share update (Herbster and Warmuth, 1998); in our efforts we found many technical problems with this approach. On both the more practical and speculative side; we observe that the specialists sets $B_n$, and $F_n$ were chosen to “prove bounds.” In practice we can use any hierarchical graph clustering algorithm to produce a complete specialist set and furthermore multiple such clus-
terings may be pooled. Such a pooled set of subgraph “motifs” could be then be used in a multi-task setting like (Koolen et al., 2012).

Acknowledgments

The authors would like to thank Dmitry Adamskiy and Stephen Pasteris for valuable discussions. In particular we would like to thank Stephen for his refinements to the proof Proposition 3. Finally both authors would like to thank Innovate UK and the EPSRC for financial support through Innovate UK grant #103971, *Graph Analytics-based Modular Machine learning and Artificial intelligence (GAMMA).*
References

D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk. A closer look at adaptive regret. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, ALT’12, pages 290–304, 2012.

J. M. Barzdin and R. V. Freivald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.

M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML ’01, pages 19–26, 2001.

O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3(Nov):363–396, 2002.

C. Bracegirdle and D. Barber. Switch-reset models: Exact and approximate inference. In *Proceedings of The Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 190–198, 2011.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

N. Cesa-Bianchi, C. Gentile, and F. Vitale. Fast and optimal prediction on a labeled tree. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pages 145–156. Omnipress, 2009.

N. Cesa-Bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. Mirror descent meets fixed share (and feels no regret). In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS ’12, pages 980–988, 2012.

N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. Random spanning trees and the prediction of weighted graphs. *Journal of Machine Learning Research*, 14(1):1251–1284, 2013.

A. Daniely, A. Gonen, and S. Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 1405–1411, 2015.

Y. Freund. Private communication, 2000. Also posted on http://www.learning-theory.org.

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

Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC ’97, pages 334–343, 1997.
A. György, T. Linder, and G. Lugosi. Tracking the best of many experts. In Proceedings of the 18th Annual Conference on Learning Theory, COLT ’05, pages 204–216, 2005.

E. Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. Electronic Colloquium on Computational Complexity (ECCC), 14(088), 2007.

M. Herbster. Tracking the best expert II. Unpublished manuscript, 1997.

M. Herbster and G. Lever. Predicting the labelling of a graph via minimum $p$-seminorm interpolation. In COLT 2009 - The 22nd Conference on Learning Theory, 2009.

M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS’06, pages 577–584, 2006.

M. Herbster and M. K. Warmuth. Tracking the best linear predictor. Journal of Machine Learning Research, 1:281–309, September 2001.

M. Herbster and M.K. Warmuth. Tracking the best expert. Machine Learning, 32(2):151–178, 1998.

M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In Proceedings of the 22nd International Conference on Machine Learning, ICML ’05, pages 305–312, 2005.

M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In Proceedings of the 21st International Conference on Neural Information Processing Systems, NIPS ’08, pages 649–656, 2008.

M. Herbster, S. Pasteris, and S. Ghosh. Online prediction at the limit of zero temperature. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS’15, pages 2935–2943, 2015a.

M. Herbster, S. Pasteris, and M. Pontil. Predicting a switching sequence of graph labelings. Journal of Machine Learning Research, 16(1):2003–2022, 2015b.

K.S. Jun, F. Orabona, S. Wright, and R. Willett. Improved strongly adaptive online learning using coin betting. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of Proceedings of Machine Learning Research, pages 943–951. PMLR, 20–22 Apr 2017.

B. S. Kerner. Experimental features of self-organization in traffic flow. Phys. Rev. Lett., 81:3797–3800, Oct 1998.

J. Kivinen, A.J. Smola, and R.C. Williamson. Online learning with kernels. Trans. Sig. Proc., 52 (8):2165–2176, August 2004.

D. J. Klein and M. Randić. Resistance distance. Journal of mathematical chemistry, 12(1):81–95, 1993.

W. M. Koolen and S. Rooij. Combining expert advice efficiently. In 21st Annual Conference on Learning Theory - COLT 2008, pages 275–286, 2008.
W. M. Koolen, D. Adamskiy, and M. K. Warmuth. Putting bayes to sleep. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS ’12, pages 135–143, 2012.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. Information and Computation, 108(2):212–261, 1994.

R. Lyons and Y. Peres. Probability on Trees and Networks. Cambridge University Press, New York, NY, USA, 1st edition, 2017. ISBN 1107160154, 9781107160156.

O. H. M. Padilla, J. Sharpnack, J. G. Scott, and R. J. Tibshirani. The dfs fused lasso: Linear-time denoising over general graphs. Journal of Machine Learning Research, 18(1):1–36, 2018.

J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. SIAM Journal on Computing, 12(4):777–788, 1983.

A. Rakhlin and K. Sridharan. Efficient online multiclass prediction on graphs via surrogate losses. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, pages 1403–1411, 2017.

F. Vitale, N. Cesa-Bianchi, C. Gentile, and G. Zappella. See the tree through the lines: The shazoo algorithm. In Advances in Neural Information Processing Systems 23, pages 1584–1592, 2011.

V. Vovk. Aggregating strategies. In Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT ’90, pages 371–386, 1990. ISBN 1-55860-146-5.

V. Vovk. Derandomizing stochastic prediction strategies. Machine Learning, 35(3):247–282, 1999.

D. B. Wilson. Generating random spanning trees more quickly than the cover time. In Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, STOC ’96, pages 296–303, 1996.

D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS ’03, pages 321–328, 2003.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML ’03, pages 912–919, 2003.
Appendix A. Proof of Theorem 1

Proof Applying the standard Halving analysis to (2) yields a bound of

\[ M_B \leq (|K| - 1) \log \frac{1}{\alpha} + (T - |K|) \log \frac{1}{1 - \alpha} + |K| + |K| \Phi \log \frac{1}{\theta} + |K| (n - 1 - \Phi) \log \frac{1}{1 - \theta}. \]  

(7)

By making the classifier conservative (see section 3), without loss of generality we assume that a mistake is made on every trial, that is, \( T = M_B \). The upper bound (7) is then minimized by setting \( \alpha = \frac{|K|}{M_B - 1} \) and \( \theta = \frac{\Phi}{n - 1} \). Substituting in these values and using \( \log_2 (1 + x) \leq \frac{x}{m^2} \) for \( x > 0 \) we obtain

\[ M_B \leq \sum_{k \in K} \Phi_k \log \frac{n - 1}{\Phi} + \frac{1}{\ln 2} \sum_{k \in K} (\Phi_k + 1) + |K| + (|K| - 1) \log \frac{M_B - 1}{|K| - 1}. \]

Now let

\[ Z := \sum_{k \in K} \Phi_k \log \frac{n - 1}{\Phi} + \frac{1}{\ln 2} \sum_{k \in K} (\Phi_k + 1) + |K| - (|K| - 1) \log (|K| - 1), \]

and \( r := (|K| - 1) \). This bound is self-referential, and thus we can iteratively substitute the bound into itself to give

\[ M_B \leq Z + r \log (M_B - 1) \]

\[ \leq Z + r \log (Z + r \log (M_B - 1)) \]

\[ \leq Z + r \log Z + r \log r + r \log \log (M_B - 1) \]  

(since \( \log (a + b) \leq \log ab \) for \( a, b \geq 2 \)

\[ = \sum_{k \in K} \Phi_k \log \frac{n - 1}{\Phi} + \frac{1}{\ln 2} \sum_{k \in K} (\Phi_k + 1) + |K| + (|K| - 1) \log Z + (|K| - 1) \log \log (T - 1), \]

and since \( (|K| - 1) \log Z \leq O(|K| \log (\sum_{k \in K} \Phi_k)) \) \( \leq O(|K| \log (|K| n)) \) we have

\[ M_B \leq O \left( \sum_{k \in K} \Phi_k \log \frac{n}{\Phi} + |K| \log |K| + |K| \log \log T \right). \]

Appendix B. Efficient calculation of the classifier \( \hat{y}_t^x \)

We first introduce some notational simplifications. Let \( D_{[s,t]} = (i_s, y_s), \ldots, (i_{t-1}, y_{t-1}) \) be the vertex-label pairs from trial \( s \) to trial \( t - 1 \). We use the abbreviation

\[ p(y_t | D_{[1,t]}) := p_S^{xy}(u_{i_t} = y_t | u_{i_1} = y_1, \ldots, u_{i_{t-1}} = y_{t-1}), \]

which in fact is the quantity we need for our prediction \( \hat{y}_t^x \). Now recall that \( S_t \) is the Bernoulli switch-reset random variable where \( S_t = 1 \) is the event that indicates that a switch occurred at the end of trial \( t \) and \( S_t = 0 \) otherwise. We define the random variable \( Z_k^t := S_k \wedge (\overline{S_{k+1}} \wedge \cdots \wedge \overline{S_t}) \)
and hence \(z_k^t\) denotes the event of a switch at the end of trial \(k\), and no switches on later trials \(k+1, \ldots, t\). For convenience we define \(S_0 = 1\).

Note that when making a prediction on trial \(t\) we have that \(p(y_t|D_{[1,t]}, z_k^t) = p(y_t|D_{(k,t)}, z_k^t)\). That is, given that the last switch occurred at the end of trial \(k\), we can ignore all data observed before trial \(k + 1\) when predicting the next label. Thus,

\[
p(y_t|D_{[1,t])} = \sum_{s=0}^{t-1} p(y_t, z_s^t|D_{[1,t)}) = \sum_{s=0}^{t-1} p(y_t|D_{[1,t)}, z_s^t)p(z_s^t|D_{[1,t)})
= \sum_{s=0}^{t-1} p(y_t|D_{(s,t)}, z_s^t)p(z_s^t|D_{[1,t)})
= \sum_{s=0}^{t-1} \frac{1}{p(D_{[1,t])}} \frac{2}{p(D_{(s,t)}|z_s^t)} \frac{3}{p(z_s^t)}.
\]

(8)

We now describe how to compute terms 1, 2, 3, 4 of (8) efficiently for \(t \geq 2\) for a fixed \(s < t\). We need not consider the first trial since for any vertex \(p(y_1 = 1) = p(y_1 = -1) = \frac{1}{2}\).

**Term 1** We observe that computing \(p(y_t|D_{(s,t)}, z_s^t)\) reduces to the problem of predicting the label of a vertex on a partially labeled spine in the standard (non-switching) online setting, where the learner only observes data from trial \(s + 1\) onwards. Predicting with this conditional probability has been shown to be equivalent to implementing the 1-nearest neighbor algorithm on the spine (Herbster et al., 2008). That is, given a partial labeling \(D_{(s,t)}\), the probability distribution over the label of vertex \(i_t\) is conditionally independent of all other labels except those of its nearest neighbors observed after trial \(s + 1\), which we denote \((u_t, u_r)\), either of which may be vacuous. We use this fact to compute \(p(y_t|D_{(s,t)}, z_s^t)\):

\[
p(y_t|D_{(s,t)}, z_s^t) = \frac{p(y_t|u_t, u_r)}{p(u_t, u_r)} = \frac{p(u_t|y_t, u_r)p(y_t|u_r)}{p(u_t|u_r)}.
\]

(9)

Now given two vertices, \(p\) and \(q\), recall that \(\theta\) is the probability of a cut occurring on any given edge, then for \(\theta < \frac{1}{2}\),

\[
p_S(u_p = y|u_q = y) = \frac{1}{2}(1 + (1 - 2\theta))|q-p|.
\]

This can be used in (9) as required, for example let \(u_t = 1, u_r = -1\), \(p(y_t = 1|D_{(s,t)}, z_s^t)\) is then given by

\[
\frac{(1 + (1 - 2\theta)^{|t-i_t|})(1 - (1 - 2\theta)^{|r-i_t|})}{2(1 - (1 - 2\theta)^{|t-r|})},
\]

which can be computed in \(O(\log n)\) time using a self-balancing (e.g., red-black) binary-search tree (Herbster et al., 2008). On trial \(t\) we are required to maintain \(t\) such self-balancing trees, creating a new empty tree and updating all existing trees on each trial, requiring \(O(t \log n)\) time on trial \(t\).
Term 2  We begin with the following decomposition:

\[ p(D_{[1,t]} | z^t_s) = p(D_{[1,s]} | z^t_s) p(D_{(s,t)} | z^t_s) \]
\[ = p(D_{[1,s]}) p(D_{(s,t)} | z^t_s) \]
\[ = p(D_{[1,s]}) p(D_{t-1} | D_{(s,t-1)}, z^t_s) p(D_{(s,t-1)} | z^t_s), \]

and see that \( p(D_{[1,s]}) \) corresponds to term 4 of (8) computed on a previous trial \( s < t \). We also observe that \( p(D_{t-1} | D_{(s,t-1)}, z^t_s) \) is simply \( p(y_{t-1} | D_{(s,t-1)}, z^t_s^{-1}) \), which corresponds to term 1 on the previous trial and finally \( p(D_{(s,t-1)} | z^t_s) = p(y_{t-2} | D_{(s,t-2)}, z^t_s) p(D_{(s,t-2)} | z^t_s) \) which again gives terms computed on the previous trial (terms 1 and 2 respectively). Term 2 can therefore be computed in \( O(1) \) time.

Term 3  Recall that \( Z'_k := S_k \land (\overline{S_{k+1}} \land \cdots \land \overline{S_t}) \), where \( \langle S_k, \ldots, S_t \rangle \) are i.i.d. Bernoulli random variables, then

\[ p(z^t_s) = \begin{cases} 
(1 - \alpha)^{t-s-1} & \text{if } s = 0, \\
\alpha (1 - \alpha)^{t-s-1} & \text{otherwise,} 
\end{cases} \]

requiring \( O(1) \) time.

Term 4  Recall that \( p(D_1) = p(y_1) = \frac{1}{2} \). We then have for \( t > 2 \),

\[ p(D_{[1,t]}) = p(D_{t-1} | D_{[1,t-1]}) p(D_{[1,t-1]}) \]
\[ = p(y_{t-1} | D_{[1,t-1]}) p(D_{[1,t-1]}), \]

where \( p(y_{t-1} | D_{[1,t-1]}) \) is simply our marginalized probability (8) calculated on the previous trial, and \( p(D_{[1,t-1]}) \) is again just term 4 on the previous trial. Thus term 4 also requires \( O(1) \) time.

Computing these terms for \( s = 0, \ldots, t-1 \) we conclude that using dynamic programming this algorithm can compute \( p(y_t | D_{[1,t]}) \) in \( O(t \log n) \) time and requires \( O(tn) \) space on trial \( t \).

Appendix C. Proof of Theorem 3

Proof  Recall that the cached share update (3) is equivalent to performing (5). We thus simulate the latter update in our analysis. Also without loss of generality we assume that a mistake is made on every trial.

We first argue the inequality

\[ [\hat{y}_t \neq y_t] \leq \frac{1}{\mu_t(Y_t)} \left( d(\mu_t, \omega_t) - d(\mu_t, \hat{\omega}_t) \right), \quad (10) \]

as this is derived by observing that

\[ d(\mu_t, \omega_t) - d(\mu_t, \hat{\omega}_t) = \sum_{\varepsilon \in E} \mu_{t,\varepsilon} \log \frac{\hat{\omega}_{t,\varepsilon}}{\omega_{t,\varepsilon}} \]
\[ = \sum_{\varepsilon \in Y_t} \mu_{t,\varepsilon} \log \frac{\hat{\omega}_{t,\varepsilon}}{\omega_{t,\varepsilon}} \]
\[ \geq \mu_t(Y_t) [\hat{y}_t \neq y_t], \]
where the second line follows the fact that \( \mu_{t,e} \log \frac{\omega_{t+1,e}}{\omega_{t,e}} = 0 \) if \( e \not\in \mathcal{Y}_t \) as either the specialist \( e \) predicts ‘\( \Box \)’ and \( \dot{\omega}_{t,e} = \omega_{t,e} \) or it predicts incorrectly and hence \( \mu_{t,e} = 0 \). The third line follows as for \( e \in \mathcal{Y}_t, \frac{\omega_{t+1,e}}{\omega_{t,e}} \geq 2 \) if there has been a mistake on trial \( t \) and otherwise the ratio is \( \geq 1 \). Indeed, since Algorithm 1 is conservative, this ratio is exactly 1 when no mistake is made on trial \( t \), thus without loss of generality we will assume the algorithm makes a mistake on every trial.

For clarity we will now use simplified notation and let \( \pi_t := \mu_t(\mathcal{Y}_t) \). We now prove the following inequalities which we will add to (10) to create a telescoping sum of relative entropy terms and entropy terms.

\[
\frac{1}{\pi_t} [d(\mu_t, \dot{\omega}_t) - d(\mu_t, \omega_{t+1})] \geq -\frac{1}{\pi_t} \log \frac{1}{1 - \alpha}, \tag{11}
\]

\[
\frac{1}{\pi_t} d(\mu_t, \omega_{t+1}) - \frac{1}{\pi_{t+1}} d(\mu_{t+1}, \omega_{t+1}) \geq -\frac{1}{\pi_t} H(\mu_t) + \frac{1}{\pi_{t+1}} H(\mu_{t+1}) - J_\varepsilon(\mu_t, \mu_{t+1}) \log \frac{|\mathcal{E}|}{\alpha}. \tag{12}
\]

Firstly (11) is proved with the following

\[
d(\mu_t, \omega_t) - d(\mu_t, \omega_{t+1}) = \sum_{e \in \mathcal{E}} \mu_{t,e} \log \frac{\omega_{t+1,e}}{\omega_{t,e}} \geq \sum_{e \in \mathcal{E}} \mu_{t,e} \log \left( \frac{(1 - \alpha)\dot{\omega}_{t,e}}{\dot{\omega}_{t,e}} \right),
\]

where the inequality has used \( \omega_{t+1,e} \geq (1 - \alpha)\dot{\omega}_{t,e} \) from (5).

To prove (12) we first define the following sets.

\[
\Theta_t := \{ e \in \mathcal{E} : \mu_{t-1,e} \neq 0, \mu_{t,e} = 0 \},
\]

\[
\Psi_t := \{ e \in \mathcal{E} : \mu_{t-1,e} \neq 0, \mu_{t,e} \neq 0 \},
\]

\[
\Omega_t := \{ e \in \mathcal{E} : \mu_{t-1,e} = 0, \mu_{t,e} \neq 0 \}.
\]

We now expand the following

\[
\frac{1}{\pi_t} d(\mu_t, \omega_{t+1}) - \frac{1}{\pi_{t+1}} d(\mu_{t+1}, \omega_{t+1})
\]

\[
= \frac{1}{\pi_t} d(\mu_t, \omega_{t+1}) - \frac{1}{\pi_t} d(\mu_{t+1}, \omega_{t+1}) + \frac{1}{\pi_t} d(\mu_{t+1}, \omega_{t+1}) - \frac{1}{\pi_{t+1}} d(\mu_{t+1}, \omega_{t+1})
\]

\[
= \frac{1}{\pi_t} \sum_{e \in \mathcal{E}} \mu_{t,e} \log \frac{\mu_{t+1,e}}{\omega_{t+1,e}} - \frac{1}{\pi_t} \sum_{e \in \mathcal{E}} \mu_{t+1,e} \log \frac{\mu_{t+1,e}}{\omega_{t+1,e}}
\]

\[
+ \frac{1}{\pi_t} \sum_{e \in \mathcal{E}} \mu_{t+1,e} \log \frac{\mu_{t+1,e}}{\omega_{t+1,e}} - \frac{1}{\pi_{t+1}} \sum_{e \in \mathcal{E}} \mu_{t+1,e} \log \frac{\mu_{t+1,e}}{\omega_{t+1,e}}
\]

\[
= -\frac{1}{\pi_t} H(\mu_t) + \frac{1}{\pi_t} H(\mu_{t+1}) + \sum_{e \in \mathcal{E}} \left( \frac{\mu_{t,e}}{\pi_t} - \frac{\mu_{t+1,e}}{\pi_t} \right) \log \frac{1}{\omega_{t+1,e}}
\]

\[
- \frac{1}{\pi_t} H(\mu_{t+1}) + \frac{1}{\pi_{t+1}} H(\mu_{t+1}) + \sum_{e \in \mathcal{E}} \left( \frac{\mu_{t+1,e}}{\pi_{t+1}} - \frac{\mu_{t+1,e}}{\pi_{t+1}} \right) \log \frac{1}{\omega_{t+1,e}}. \tag{13}
\]

Recall that a comparator \( \mu \in \Delta_{|\mathcal{E}|} \) is well-formed if \( \forall v \in V \), there exists a unique \( e \in \mathcal{E} \) such that \( e(v) \neq \Box \) and \( \mu_e > 0 \), and furthermore there exists a \( \pi \in (0, 1] \) such that \( \forall e \in \mathcal{E} : \mu_e \in \{0, \pi\} \).
i.e., each specialist in the support of \( \mu \) has the same mass \( \pi \) and these specialists disjointly cover the input space \((V)\). Thus, by collecting terms into the three sets \( \Theta_{t+1}, \Psi_{t+1}, \) and \( \Omega_{t+1} \) we have

\[
\sum_{e \in \mathcal{E}} \left( \frac{\mu_{t,e}}{\pi_t} - \frac{\mu_{t+1,e}}{\pi_t} \right) \log \frac{1}{\omega_{t+1,e}} = \sum_{e \in \Theta_{t+1}} \frac{\mu_{t,e}}{\pi_t} \log \frac{1}{\omega_{t+1,e}} + \sum_{e \in \Psi_{t+1}} \left( \frac{\mu_{t,e}}{\pi_t} - \frac{\mu_{t+1,e}}{\pi_t} \right) \log \frac{1}{\omega_{t+1,e}} - \sum_{e \in \Omega_{t+1}} \frac{\mu_{t+1,e}}{\pi_t} \log \frac{1}{\omega_{t+1,e}}
\]

and similarly

\[
\sum_{e \in \mathcal{E}} \left( \frac{\mu_{t+1,e}}{\pi_t} - \frac{\mu_{t+1,e}}{\pi_{t+1}} \right) \log \frac{1}{\omega_{t+1,e}} = \sum_{e \in \Psi_{t+1}} \left( \frac{\mu_{t+1,e}}{\pi_t} - 1 \right) \log \frac{1}{\omega_{t+1,e}} + \sum_{e \in \Omega_{t+1}} \left( \frac{\mu_{t+1,e}}{\pi_t} - 1 \right) \log \frac{1}{\omega_{t+1,e}}.
\]

Substituting (14) and (15) into (13) and simplifying gives

\[
\frac{1}{\pi_t} d(\mu_t, \omega_{t+1}) - \frac{1}{\pi_{t+1}} d(\mu_{t+1}, \omega_{t+1}) = -\frac{1}{\pi_t} H(\mu_t) + \frac{1}{\pi_{t+1}} H(\mu_{t+1}) + \sum_{e \in \Theta_{t+1}} \frac{\mu_{t,e}}{\pi_t} \log \frac{1}{\omega_{t+1,e}} - \sum_{e \in \Omega_{t+1}} \log \frac{1}{\omega_{t+1,e}} \\
\geq -\frac{1}{\pi_t} H(\mu_t) + \frac{1}{\pi_{t+1}} H(\mu_{t+1}) - |\Omega_{t+1}| \log \frac{\mathcal{E}}{\alpha},
\]

where the inequality has used the fact that \( \frac{\alpha}{|\mathcal{E}|} \leq \omega_{t+1,e} \leq 1 \) from (5).

Summing over all trials then leaves a telescoping sum of relative entropy terms, a cost of \( \frac{1}{\pi_t} \log \frac{1}{1-\alpha} \) on each trial, and \( |\Omega_{t+1}| \log \frac{\mathcal{E}}{\alpha} \) for each switch. Thus,

\[
\sum_{t=1}^{T} \left[ y_t \neq y_t \right] \leq \frac{1}{\pi_1} d(\mu_1, \omega_1) + \frac{1}{\pi_1} H(\mu_1) + \sum_{t=1}^{T} \frac{1}{\pi_t} \log \frac{1}{1-\alpha} + \sum_{i=1}^{K-1} J_{\mathcal{E}} (\mu_{k_i}, \mu_{k_{i+1}}) \log \frac{\mathcal{E}}{\alpha},
\]

where \( J_{\mathcal{E}} (\mu_{k_i}, \mu_{k_{i+1}}) = |\Omega_{k_{i+1}}| \), and since \( \omega_1 = \frac{1}{|\mathcal{E}|} \mathbf{1} \), we can combine the remaining entropy and relative entropy terms to give \( \frac{1}{\pi_1} d(\mu_1, \omega_1) + \frac{1}{\pi_1} H(\mu_1) = \frac{1}{\pi_1} \log |\mathcal{E}| \), concluding the proof. \( \blacksquare \)

**Appendix D. Proof of Proposition 4**

We recall the proposition:
The basis $\mathcal{B}_n$ is complete. Furthermore, for any labeling $u \in \{-1, 1\}^n$ there exists a covering set $C_u \subseteq \mathcal{B}_n$ such that $|C_u| \leq 2(\Phi_S(u) + 1)\lfloor \log_2 \frac{n}{2} \rfloor$.

We first give a brief intuition of the proof; any required terms will be defined more completely later. For a given labeling $u \in \{-1, 1\}^n$ of cut-size $\Phi_S(u)$, the spine $S$ can be cut into $\Phi_S(u) + 1$ clusters, where a cluster is a contiguous segment of vertices with the same label. We will upper bound the maximum number of cluster specialists required to cover a single cluster, and therefore obtain an upper bound for $|C_u|$ by summing over the $\Phi_S(u) + 1$ clusters.

Without loss of generality we assume $n = 2^r$ for some integer $r$ and thus the structure of $\mathcal{B}_n$ is analogous to a binary tree of depth $d = \log_2 n$. Indeed, for a fixed label parameter $y$ we will adopt the terminology of binary trees such that for instance we say specialist $\varepsilon_{i,j}^{l,r}$ for $i \neq j$ has a so-called left-child $\varepsilon_y^{i,\lfloor \frac{i+j}{2} \rfloor}$ and right-child $\varepsilon_y^{\lceil \frac{i+j}{2} \rceil,j}$. Similarly, we say that $\varepsilon_y^{i,\lfloor \frac{i+j}{2} \rfloor}$ and $\varepsilon_y^{\lceil \frac{i+j}{2} \rceil,j}$ are siblings, and $\varepsilon_y^{i,j}$ is their parent. Note that any specialist is both an ancestor and a descendant of itself, and a proper descendant of a specialist is a descendant of one of its children. Finally the depth of specialist $\varepsilon_{i,j}^{l,r}$ is defined to be equal to the depth of the corresponding node in a binary tree, such that $\varepsilon_y^{1,n}$ is of depth $0$, $\varepsilon_y^{1,2}$ and $\varepsilon_y^{n+1,2}$ are of depth $1$, etc.

The first claim of the proposition is easy to prove as $\{\varepsilon_y^{i,1}, \varepsilon_y^{1,j} : i \in [n]\} \subseteq \mathcal{B}_n$ and thus any labeling $u \in \{-1, 1\}^n$ can be covered. We now prove the second claim of the proposition.

We will denote a uniformly-labeled contiguous segment of vertices by the pair $(l, r)$, where $l, r \in [n]$ are the two end vertices of the segment. For completeness we will allow the trivial case when $l = r$. Given a labeling $u \in \{-1, 1\}^n$, let $\mathcal{L}_u := \{(l, r) : 1 \leq l \leq r \leq n; u_l = \ldots = u_r; u_{l-1} \neq u_l; u_{r+1} \neq u_r\}$ be the set of maximum-sized contiguous segments of uniformly-labeled vertices. Note that $u_{l-1}$ or $u_{r+1}$ may be vacuous. When the context is clear, we will also describe $(l, r)$ as a cluster, and as the set of vertices $\{l, \ldots, r\}$.

For a given $u \in \{-1, 1\}^n$ and cluster $(l, r) \in \mathcal{L}_u$, we say $\mathcal{B}_{(l,r)} \subseteq \mathcal{B}_n$ is an $(l, r)$-covering set with respect to $u$ if for all $\varepsilon_y^{i,j} \in \mathcal{B}_{(l,r)}$ we have $l \leq i, j \leq r$, and if for all $k \in (l, r)$ there exists some $\varepsilon_y^{i,j} \in \mathcal{B}_{(l,r)}$ such that $i \leq k \leq j$ and $y = u_k$. That is, every vertex in the cluster is ‘covered’ by at least one specialist and no specialists cover any vertices $k \notin (l, r)$. We define $\mathcal{D}^{(l,r)}$ to be the set of all possible $(l, r)$-covering sets with respect to $u$.

We now define
\[
\delta(\mathcal{B}_{(l,r)}) := |\mathcal{B}_{(l,r)}|
\]
to be the complexity of $\mathcal{B}_{(l,r)} \in \mathcal{D}^{(l,r)}$.

For a given $u \in \{-1, 1\}^n$ and cluster $(l, r) \in \mathcal{L}_u$, we wish to produce an $(l, r)$-covering set of minimum complexity, which we denote $\mathcal{B}_{(l,r)}^* = \arg\min_{\mathcal{B}_{(l,r)} \in \mathcal{D}^{(l,r)}} \delta(\mathcal{B}_{(l,r)})$. Note that an $(l, r)$-covering set of minimum complexity cannot contain any two specialists which are siblings, since they can be removed from the set and replaced by their parent specialist.

**Lemma 9** For any $u \in \{-1, 1\}^n$, for any $(l, r) \in \mathcal{L}_u$, the $(l, r)$-covering set of minimum complexity, $\mathcal{B}_{(l,r)}^*$ contains at most two specialists of each unique depth.

**Proof** We first give an intuitive sketch of the proof. For a given $u \in \{-1, 1\}^n$ and cluster $(l, r) \in \mathcal{L}_u$ assume that there are at least three specialists of equal depth in $\mathcal{B}_{(l,r)}^*$, then any of these specialists that are in the ‘middle’ may be removed, along with any of their siblings or proper descendants that
are also members of $B^*_u$ without creating any ‘holes’ in the covering, decreasing the complexity of $B^*_{(l,r)}$.

We use a proof by contradiction. Suppose for contradiction that for a given $u \in \{-1, 1\}^n$ and $(l, r) \in L_u$, the $(l, r)$-covering set of minimum complexity, $B^*_{(l,r)}$, contains three distinct specialists of the same depth, $\varepsilon_y^{a,b}, \varepsilon_y^{c,d}, \varepsilon_y^{e,f}$. Without loss of generality let $a, b < c, d < e, f$. Note that we have $l \leq a < f \leq r$. We consider the following two possible scenarios: when two of the three specialists are siblings, and when none are.

If $\varepsilon_y^{a,b}$ and $\varepsilon_y^{c,d}$ are siblings, then we have $\varepsilon_y^{a,d} \in B_y$ and thus $\{\varepsilon_y^{a,d}\} \cup B^*_{(l,r)} \setminus \{\varepsilon_y^{a,b}, \varepsilon_y^{c,d}\}$ is an $(l, r)$-covering set of smaller complexity, leading to a contradiction. The equivalent argument holds if $\varepsilon_y^{c,d}$ and $\varepsilon_y^{e,f}$ are siblings.

If none are siblings, then let $\varepsilon_y^{c',d'}$ be the sibling of $\varepsilon_y^{c,d}$ and let $\varepsilon_y^{C,D}$ be the parent of $\varepsilon_y^{c,d}$ and $\varepsilon_y^{c',d'}$. Note that $a, b < c', d', c, d$ and $c', d', c, d < e, f$ and hence $l < C < D < r$. If an ancestor of $\varepsilon_y^{C,D}$ is in $B^*_{(l,r)}$, then $B^*_{(l,r)} \setminus \{\varepsilon_y^{c,d}\}$ is an $(l, r)$-covering set of smaller complexity, leading to a contradiction. Alternatively, if no ancestor of $\varepsilon_y^{C,D}$ is in $B^*_{(l,r)}$, then $\varepsilon_y^{c,d}$ or some of its proper descendants must be in $B^*_{(l,r)}$, otherwise there exists some vertex $k' \in (c', d')$ such that there exists no specialist $\varepsilon_y^{i,j} \in B^*_{(l,r)}$ such that $i \leq k' \leq j$, and therefore $B^*_{(l,r)}$ would not be an $(l, r)$-covering set. Let $\varepsilon_y^{p,q}$ be a descendant of $\varepsilon_y^{c',d'}$ which is contained in $B^*_{(l,r)}$. Then $\{\varepsilon_y^{C,D}\} \cup B^*_{(l,r)} \setminus \{\varepsilon_y^{c,d}, \varepsilon_y^{p,q}\}$ is an $(l, r)$-covering set of smaller complexity, leading to a contradiction.

We conclude that there can be no more than 2 specialists of the same depth in $B^*_{(l,r)}$ for any $u \in \{-1, 1\}^n$ and any $(l, r) \in L_u$. 

We now prove an upper bound on the maximum minimum-complexity of an $(l, r)$-covering set under any labeling $u$.

**Corollary 10** For all $u \in \{-1, 1\}^n$,

$$\max_{(l, r) \in L_u} \min_{B^*_{(l,r)} \in B(l,r)} \delta(B^*_{(l,r)}) \leq 2 \log \frac{n}{2}. \quad (18)$$

**Proof** For any $u \in \{-1, 1\}^n$, and $(l, r) \in L_u$, since $B^*_{(l,r)}$ can contain at most 2 specialists of the same depth (Lemma 9) an $(l, r)$-covering set of minimum-complexity can have at most two specialists of depths 2, 3, \ldots, $d$. This set cannot contain two specialists of depth 1 as they are siblings. This upper bounds the maximum minimum-complexity of any $(l, r)$-covering set by $2(d - 2) = 2 \log \frac{n}{2}$.

Finally we conclude that for any labeling $u \in \{-1, 1\}^n$ of cut-size $\Phi_S(u)$, there exists $C_u \subseteq B_n$ such that $|C_u| \leq 2 \log_2 \left(\frac{n}{2}\right) (\Phi_S(u) + 1)$.

**Appendix E. Proof of Proposition 5**

First recall the proposition statement.

**Proposition 5** For a linearized graph $S$, for comparators $\mu, \mu' \in \Delta_{|F_n|}$ that are minimal-consistent with $u$ and $u'$ respectively,

$$J_{F_n}(\mu, \mu') \leq \min \left(2H(u, u'), \Phi_S(u') + 1 \right).$$
**Proof** We prove both inequalities separately. We first prove \( J_{\mathcal{F}_n}(\mu, \mu') \leq \Phi_S(u') + 1 \). This follows directly from the fact that \( J_{\mathcal{E}}(\mu, \mu') := |\{ \varepsilon \in \mathcal{E} : \mu_\varepsilon = 0, \mu'_\varepsilon \neq 0 \}| \) and therefore \( J_{\mathcal{F}_n}(\mu, \mu') \leq |\{ \varepsilon \in \mathcal{F}_n : \mu_\varepsilon \neq 0 \}| = \Phi_S(u') + 1 \).

We now prove \( J_{\mathcal{F}_n}(\mu, \mu') \leq 2H(u, u') \). Recall that if \( u \neq u' \) then by definition of the minimal-consistent comparators \( \mu \) and \( \mu' \), the set \( \{ \varepsilon \in \mathcal{F}_n : \mu_\varepsilon = 0, \mu'_\varepsilon \neq 0 \} \) corresponds to the set of maximum-sized contiguous segments of vertices in \( S \) sharing the same label that did not exist in the labeling \( u \). From here on we refer to a maximum-sized contiguous segment as just a contiguous segment.

When switching from labeling \( u \) to \( u' \), we consider the following three cases. First when a non-cut edge (with respect to \( u \)) becomes a cut edge (with respect to \( u' \)), second when a cut edge (with respect to \( u \)) becomes a non-cut edge (with respect to \( u' \)), and lastly when a cut edge remains a cut edge, but the labeling of the two corresponding vertices are ‘swapped’.

Formally then, for an edge \((i, j) \in E_S \) such that \([u_i = u_j] \land [u'_i \neq u'_j] \) there exists two new contiguous segments of vertices sharing the same label that did not exist in the labeling \( u \), their boundary being the edge \((i, j) \).

Conversely for an edge \((i, j) \in E_S \) such that \([u_i \neq u_j] \land [u'_i = u'_j] \) there exists one new contiguous segment of vertices sharing the same label that did not exist in the labeling \( u \), that segment will contain the edge \((i, j) \).

Finally for an edge \((i, j) \in E_S \) such that \(([u_i \neq u_j] \land [u'_i \neq u'_j]) \land ([u_i \neq u'_i] \lor [u_j \neq u'_j]) \) there exists two new contiguous segments of vertices sharing the same label that did not exist in the labeling \( u \), their boundary being the edge \((i, j) \).

We conclude that the number of new contiguous segments of vertices sharing the same label that did not exist in the labeling \( u \) is upper bounded by

\[
2 \sum_{(i, j) \in E_S} \left[ [u_i \neq u_j] \lor [u'_i \neq u'_j] \right] \land \left[ [u_i \neq u'_i] \lor [u_j \neq u'_j] \right].
\]

\[
\text{(1)}
\]

**Appendix F. Proof of Corollary 6**

First recall the corollary statement.

**Corollary 6** For a connected \( n \)-vertex graph \( G \) and with randomly sampled spine \( S \), the number of mistakes made in predicting the online sequence \((i_1, y_1), \ldots, (i_T, y_T)\) by the SCS algorithm with optimally-tuned \( \alpha \) is upper bounded with basis \( \mathcal{F}_n \) by

\[
\mathcal{O} \left( \Phi_1 \log n + \sum_{i=1}^{|K|-1} H(u_{k_i}, u_{k_{i+1}}) (\log n + \log |K| + \log \log T) \right)
\]

and with basis \( \mathcal{B}_n \) by

\[
\mathcal{O} \left( \left( \Phi_1 \log n + \sum_{i=1}^{\left| \frac{|K|-1}{2} \right|} H(u_{k_i}, u_{k_{i+1}}) (\log n + \log |K| + \log \log T) \right) \log n \right)
\]

for any sequence of labelings \( u_1, \ldots, u_T \in \{-1, 1\}^n \) such that \( u_{t_i, t_i} = y_t \) for all \( t \in [T] \).
Proof  Since Algorithm 1 has a conservative update, we may ignore trials on which no mistake is made and thus from the point of view of the algorithm a mistake is made on every trial, we will therefore assume that \( T = M \). This will lead to a self-referential mistake bound in terms of the number of mistakes made which we will then iteratively substitute into itself.

Let \( c := \log_2 e \), we will use the fact that \( \log_2 \left( \frac{1}{x+y} \right) \leq c \frac{x}{y} \) for \( x, y > 0 \). We will first optimally tune \( \alpha \) to give our tuned mistake bound for a general basis set \( \mathcal{E} \), and then derive the bounds for SCS-F and SCS-B respectively.

The value of \( \alpha \) that minimizes (6) is

\[
\alpha = \frac{\sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{E}}(\mu_i, \mu_{i+1})}{\sum_{i=1}^{T} \frac{1}{\pi_i} + \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{E}}(\mu_i, \mu_{i+1})}, \tag{19}
\]

which when substituted into the second term of (6) gives

\[
M_{\mathcal{E}} \leq \frac{1}{\pi_1} \log |\mathcal{E}| + c \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{E}}(\mu_i, \mu_{i+1}) + \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{E}}(\mu_i, \mu_{i+1}) \log \frac{|\mathcal{E}|}{\alpha}. \tag{20}
\]

We now upper bound \( \frac{1}{\alpha} \) for substitution in the last term of (20) for the two bases, \( \mathcal{F}_n \) and \( \mathcal{B}_n \) separately.

**Basis \( \mathcal{F}_n \):** For \( \mathcal{F}_n \) observe that \( |\mathcal{E}| = n^2 + n \), and since any labeling \( \mu_i \in \{-1, 1\}^n \) of cut-size \( \Phi_S(\mu_i) \) is covered by \( \Phi_S(\mu_i) + 1 \) specialists, we have that \( \pi_t = 1/(\Phi_S(\mu_i) + 1) \) on all trials. We let the number of mistakes made be denoted by \( M_{\mathcal{F}_n} \). Thus (20) immediately becomes

\[
M_{\mathcal{F}_n} \leq (\Phi_1 + 1) \log |\mathcal{F}_n| + c \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{F}_n}(\mu_i, \mu_{i+1}) + \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{F}_n}(\mu_i, \mu_{i+1}) \log \frac{|\mathcal{F}_n|}{\alpha}. \tag{21}
\]

To upper bound \( \frac{1}{\alpha} \) we note that if \( \mu_i \neq \mu_{i+1} \) then \( J_{\mathcal{F}_n}(\mu_i, \mu_{i+1}) \geq 1 \), and that for \( \mathcal{F}_n \), \( \frac{1}{\pi_t} = \Phi_k + 1 \leq n \), thus from (19) we have

\[
\frac{1}{\alpha} = 1 + \frac{\sum_{i=1}^{T} \frac{1}{\pi_t} J_{\mathcal{F}_n}(\mu_i, \mu_{i+1})}{\left\lfloor \frac{K}{\pi_t} \right\rfloor - 1} \leq 1 + \frac{nT}{|K| - 1} \leq \frac{nT + |K| - 1}{|K| - 1} \leq \frac{(n+1)T}{|K| - 1}.
\]

Substituting \( \frac{1}{\alpha} \leq \frac{(n+1)T}{|K| - 1} \) into (21) gives

\[
M_{\mathcal{F}_n} \leq (\Phi_1 + 1) \log |\mathcal{F}_n| + \sum_{i=1}^{\left\lfloor \frac{K}{\pi_t} \right\rfloor} J_{\mathcal{F}_n}(\mu_i, \mu_{i+1}) \left[ \log (e|\mathcal{F}_n|) + \log (n+1) + \log \frac{T}{|K| - 1} \right]. \tag{22}
\]
We now show our method to reduce the $\log T$ term in our bound to $\log \log T$. The technique is the same as that in the proof of Theorem 1. We first simplify (22) and substitute $T = M_{F_n}$,

$$
M_{F_n} \leq (\Phi_1 + 1) \log |F_n| + \frac{|K|-1}{2} \log (e|F_n|(n+1) / |K| - 1) + \frac{|K|-1}{2} \log (M_{F_n})
$$

$$
\leq Z + \mathcal{J} \log (Z + \mathcal{J} \log M_{F_n})
$$

$$
\leq Z + \mathcal{J} \log Z + \mathcal{J} \log \log M_{F_n},
$$

using $\log (a+b) \leq \log (a) + \log (b)$ for $a, b \geq 2$. We finally use the fact that $\mathcal{J} = O(n|K|)$ to give $\mathcal{J} \log \mathcal{J} \leq O(\mathcal{J} \log (n|K|))$ and similarly

$$
\mathcal{J} \log Z \leq O(\mathcal{J} \log (\Phi_1 \log n + \mathcal{J} \log n))
$$

$$
\leq O(\mathcal{J} \log ((n + \mathcal{J}) \log n))
$$

$$
\leq O(\mathcal{J} \log (n + \mathcal{J}))
$$

$$
\leq O(\mathcal{J} \log (n|K|)),
$$

to give

$$
M_{F_n} \leq O \left( \Phi_1 \log n + \sum_{i=1}^{[K]-1} J_{F_n} (\mu_{k_i}, \mu_{k_i+1}) (\log n + \log |K| + \log \log T) \right).
$$

**Basis $B_n$:** For $B_n$ we apply the same technique as above, but first observe the following. Without loss of generality assume $n = 2^r$ for some integer $r$, we then have $|E| = 4n - 2$. We let the number of mistakes made by SCS-B be denoted by $M_{B_n}$. Thus for basis $B_n$ (20) becomes

$$
M_{B_n} \leq 2 \log \frac{n}{2} (\Phi_1 + 1) \log |B_n| + c \sum_{i=1}^{[K]-1} J_{B_n} (\mu_{k_i}, \mu_{k_i+1}) + \sum_{i=1}^{[K]-1} J_{B_n} (\mu_{k_i}, \mu_{k_i+1}) \log \frac{|B_n|}{\alpha}.
$$

Recall proposition 4 (that $|C_u| \leq 2 \log_2 \left( \frac{n}{2} (\Phi_S (u) + 1) \right)$ and since $\pi_t = \frac{1}{|C_u|}$, then for any labeling $u_t \in \{-1, 1\}^n$ of cut-size $\Phi_S (u_t)$ we have $\frac{1}{2(|\Phi_S (u_t) + 1| \log n)} \leq \pi_t \leq \frac{1}{\Phi_S (u_t) + 1}$. We then apply the same argument upper bounding $\frac{1}{\alpha}$,

$$
1 - \frac{1}{\alpha} = 1 + \frac{\sum_{i=1}^{[K]-1} J_{B_n} (\mu_{k_i}, \mu_{k_i+1})}{\pi_t} \leq 1 + \frac{2n \log \left( \frac{n}{2} \right) T}{|K| - 1} \leq \frac{2n \log \left( \frac{n}{2} \right) T + |K| - 1}{|K| - 1} \leq \frac{2n \log \left( \frac{n}{2} \right) T + |K| - 1}{|K| - 1},
$$

and subbing $\frac{1}{\alpha} \leq \frac{(2n \log \left( \frac{n}{2} \right) T + |K| - 1)}{|K| - 1}$ into the last term of (23) gives

$$
M_{B_n} \leq 2 \log \frac{n}{2} (\Phi_1 + 1) \log |B_n| +
$$

$$
\sum_{i=1}^{[K]-1} J_{B_n} (\mu_{k_i}, \mu_{k_i+1}) \left[ c + \log |B_n| + \ln 2n + \log \frac{T}{|K| - 1} + \log \log n \right].
$$
Applying the same recursive technique as above yields a bound of
\[ M_{B_n} \leq O \left( \Phi_1 (\log n)^2 + \sum_{i=1}^{\lfloor K \rfloor} J_{B_n}(\mu_{k_i}, \mu_{k_{i+1}}) (\log n + \log \log \log T) \right). \]

Using the same argument given in proposition 4 for any two labelings \( u, u' \in \{-1, 1\}^n \), for two consistent well-formed comparators \( \mu, \mu' \in \Delta_{|B_n|} \) respectively, and for two consistent well-formed comparators \( \hat{\mu}, \hat{\mu}' \in \Delta_{|F_n|} \), we have that \( J_{B_n}(\mu, \mu') \leq 2 \log \frac{\mu}{2} J_{F_n}(\hat{\mu}, \hat{\mu}') \). Finally we use \( J_{F_n} \leq 2H(u, u') \) from Proposition 5 to complete the proof.

**Appendix G. Proof of Proposition 7**

**Proof** Using a time-dependent \( \alpha \) we can re-write (6) as
\[ M_\mathcal{E} \leq \frac{1}{\pi_1} \log |\mathcal{E}| + \sum_{t=1}^{T} \frac{1}{\pi_t} \log \left( \frac{1}{1 - \frac{1}{t+1}} \right) + \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \log (|\mathcal{E}| (k_{i+1} + 1)) \]
and letting \( \alpha_t := \frac{1}{t+1} \), and letting \( c := \log_2 c \), gives the following,
\[ M_\mathcal{E} \leq \frac{1}{\pi_1} \log |\mathcal{E}| + \sum_{t=1}^{T} \frac{1}{\pi_t} \log \left( \frac{1}{t} \right) + \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \log (|\mathcal{E}|T) \]
\[ \leq \frac{1}{\pi_1} \log |\mathcal{E}| + c \left( \max_{t \in [T]} \frac{1}{\pi_t} \right) \sum_{t=1}^{T} \frac{1}{t} + \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \log (|\mathcal{E}|T) \]
\[ \leq \frac{1}{\pi_1} \log |\mathcal{E}| + \left( \max_{t \in [T]} \frac{1}{\pi_t} \right) \log (eT) + \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \log (|\mathcal{E}|T) \]
where the step from (25) to (26) has used \( \log_2 (1 + x) \leq cx \) for \( x > 0 \), and the step from (27) to (28) has used \( \sum_{t \in [T]} \frac{1}{t} < \int_1^T \frac{1}{t} dt + 1 = \ln (eT) = \frac{1}{c} \log_2 (eT) \).

We now use the following upper bound on \( \max_{t \in [T]} \frac{1}{\pi_t} \),
\[ \max_{t \in [T]} \frac{1}{\pi_t} \leq \frac{1}{\pi_1} + \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \]
and the assumption that \( \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}) \geq \frac{1}{\pi_1} \), to give
\[ \max_{t \in [T]} \frac{1}{\pi_t} \leq 2 \sum_{i=1}^{\lfloor K \rfloor} J_\mathcal{E}(\mu_{k_i}, \mu_{k_{i+1}}). \]
Substituting (29) into (28) then gives

\[ M_\mathcal{E} \leq \frac{1}{\pi_1} \log |\mathcal{E}| + 2 \sum_{i=1}^{\lfloor K \rfloor - 1} J_\mathcal{E}(\boldsymbol{\mu}_{k_i}, \boldsymbol{\mu}_{k_i+1}) \left( \log (eT) + \frac{1}{2} \log (|\mathcal{E}|) \right) \]

\[ = \frac{1}{\pi_1} \log |\mathcal{E}| + 2 \sum_{i=1}^{\lfloor K \rfloor - 1} J_\mathcal{E}(\boldsymbol{\mu}_{k_i}, \boldsymbol{\mu}_{k_i+1}) \left( \frac{1}{2} \log (|\mathcal{E}|) + \log (e) + \frac{3}{2} \log (T) \right) \]

Using a conservative update (see section 4.1), we similarly set \( \alpha_t := \frac{1}{m+1} \), where \( m \) is the current number of mistakes of the algorithm. We next use the same ‘recursive trick’ as that in the proof of Corollary 6. The proof follows analogously, leaving

\[ M_{\mathcal{F}_n} \leq O \left( \Phi_1 \log n + \sum_{i=1}^{\lfloor K \rfloor - 1} J_{\mathcal{F}_n}(\boldsymbol{\mu}_{k_i}, \boldsymbol{\mu}_{k_i+1}) \left( \log n + \log |K| + \log \log T \right) \right) \]

for the basis set \( \mathcal{F}_n \), and

\[ M_{\mathcal{B}_n} \leq O \left( \Phi_1 (\log n)^2 + \sum_{i=1}^{\lfloor K \rfloor - 1} J_{\mathcal{B}_n}(\boldsymbol{\mu}_{k_i}, \boldsymbol{\mu}_{k_i+1}) \left( \log n + \log |K| + \log \log T \right) \right) \]

for the basis set \( \mathcal{B}_n \).
Appendix H. The Switching Graph Perceptron

Algorithm 2: Switching Graph Perceptron

\begin{algorithm}
\begin{algorithmic}
\State \textbf{input} : Graph $G$
\State \textbf{parameter} : $\gamma > 0$
\State \textbf{initialize} : $w_1 \leftarrow 0$
\State $K \leftarrow L_G^+ + \max_{i \in [n]}(e_i^T L_G^+ e_i)11^T$
\For{$t = 1$ to $T$}
\State receive $i_t \in V$
\State predict $\hat{y}_t \leftarrow \text{sign}(w_{t,i_t})$
\If{$\hat{y}_t \neq y_t$}
\State $\hat{w}_t \leftarrow w_t + y_t \frac{K e_{i_t}}{K_{i_t,i_t}}$
\If{$\|\hat{w}_t\|_K > \gamma$}
\State $w_{t+1} \leftarrow \frac{\hat{w}_t}{\|\hat{w}_t\|_K} \gamma$
\Else
\State $w_{t+1} \leftarrow \hat{w}_t$
\EndIf
\Else
\State $w_{t+1} \leftarrow w_t$
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

In this section for completeness we provide the switching graph perceptron algorithm (SGP). The algorithm is described and a mistake bound given for the switching-graph labeling problem in (Herbster et al., 2015b, Sec. 6.2).

The key to the approach is to use the following graph kernel (introduced by Herbster and Pontil 2006) $K := L_G^+ + R_L 11^T$ with $R_L := \max_{i \in [n]}(e_i^T L_G^+ e_i)$, where $L_G^+$ denotes the pseudo-inverse of the graph Laplacian, and for $i \in [n]$, we let $e_i$ denote the $i$-th unit basis vector, i.e., $e_{i,i'} = 0$ if $i \neq i'$ and equals 1 if $i' = i$. The norm induced by this kernel is denoted $\|u\|_K := \sqrt{u^T K^{-1} u}$.

Appendix I. Proof of Proposition 8

Proof Recall that for $r$ algorithms with mistake bounds $M_1, \ldots, M_r$, a mistake on a given trial means that at least $r/2$ algorithms incurred a mistake. Since the ensemble members can incur a total of $\sum_{i=1}^r M_i$ mistakes, the mistake bound is then $2(\sum_{i=1}^r M_i)/r$. ■

29