ECE 543 Final Project: Multi-armed Bandit and Its Application to Boosting

Liming Wang lwang114@illinois.edu
Department of Electrical and Computer Engineering
University of Illinois
Urbana, IL 61820, USA

Editor:

Abstract

Keywords: Multi-armed bandit, Adaboost, multi-class classification, online learning

1. Introduction

Classical Adaboost algorithm and its multiclass generalizations (Mukherjee and Schapire (2013)) have been successfully applied to problems such as object detection (Viola and Jones (2002)), name entity recognition and text category classification (Schapire and Singer). The algorithm seeks to build a strong classifier from weak ones by greedily computing a weighted votes of the set of weak classifiers. While enjoying theoretical guarantees on training and generalization, the algorithm is, however, not online and can be time consuming as it requires passing through all training examples at each iteration.

On the other hand, the multi-armed bandit (MAB) algorithm has been successful in an online, partially observed settings, e.g., crowdsourcing (Zhou et al. (2014)), game playing and anomaly detection. This report presents the recent attempts (Jung et al. (2017); Zhang et al. (2019); Fekete and Kegl (2009)) to incorporate bandit-based sampling to feature selection, expert selection and cost matrix estimation of online Adaboost algorithm. The report is organized as follows: Section two formulates the problem of MAB and Adaboost, and section three will analyze algorithms for MAB and the general framework for the multiclass Adaboost problem proposed by Mukherjee and Schapire (2013); section four will discuss the application of MAB to Adaboost; Section five will be conclusion and future works.

2. Problem Formulation

For the MAB problem, suppose there is a bandit with K arm and each arm will generate a random reward with mean $\mu_i, i \in [K]$, where $[N]$ stands for integer from 1 to N. At each time step t, a player pulls one arm $i_t \in [K]$ and receives a reward $R_{i_t,t}$. The goal of the player is to minimize her *regret*: $\bar{R}_T((i_t)) := \mathbb{E}\left[\max_{i\in[K]} \sum_{t=1}^{T} R_{i,t} - \sum_{t=1}^{T} R_{i_t,t}\right]$. In addition, the player may want to find the *best* arm as quickly as possible in T rounds, and

the problem can be formulated as:

$$\min T = \min \sum_{i=1}^{K} T_i,$$
$$\text{s.t.} \quad \mathbb{P}\left\{\mu_{i_T} \geq \max_i \mu_i - \epsilon\right\} \geq 1 - \delta,$$

where T_i is the number of times when arm i is pulled.

The problem can be formulated alternatively in an adversarial setting, where an adversary with control over the bandit machines chooses a reward at each time for the arm pulled by the player to maximize the regret, while the learner is trying to minimize the regret: $R_T^a((i_t)) := \min_{(i_t)} \max_i \sum_{t=1}^{T} r_{i,t} - \sum_{t=1}^{T} r_{i_t,t}$.

On the other hand, the multi-class boosting problem seeks to build a strong multi-class classifier from a set of weak multi-lass classifiers $h_i \in \mathcal{H}, i = 1, \ldots, [N]$. The problem of feature selection for Adaboost can be formulated as an *exploration* problem with bandit feedback. This setting assumes that $R_{i,t}$ is time-invariant, i.e., $\theta_{i,t} =: \theta_i, \forall t$. As a result, one optimal policy will trivially be choosing the best arm, or in general, best K arms with the highest expected reward $\widehat{u} = f_t(H)$ after T trials for some T. At each iteration t, an adversary chooses data x_t, and the strong classifier outputs a prediction $\tilde{y}_t$ and receives a zero-one loss as feedback: $X_{i,t} = \mathbf{1}[\tilde{y}_t = y_t]$, where y_t is the true label at time t.

3. Algorithms and Analysis

3.1 Upper bounds on the performance of optimal multi-arm bandit algorithm

Following Katselis, an insightful way to rewrite the regret function is as follows:

$$\begin{aligned} R_T((i_t)) &= \mathbb{E}\left[\sum_{t=1}^{T} R_{i_t,t} - \inf_i R_{i,t}\right] = \mathbb{E}\left[\sum_{i=1}^{K} (R_{i,t} - R_{i^*,t}) T_i\right] \\ &= \sum_{i=1}^{K} (\mu_{i,t} - \mu_{i^*,t}) \mathbb{E}\left[T_i\right] = \sum_{i=1}^{K} \Delta_i \mathbb{E}\left[T_i\right]. \end{aligned} \tag{1}$$

As a result, upper bounding the regret amounts to upper bound the expected sampling time for arm i, $\mathbb{E}\left[T_i\right]$.

Theorem 1 *Let $\alpha_t > 2$, then*

$$R_T((i_t^{UCB})) \leq \left(\sum_{i:\Delta_i > 0} \frac{2\alpha}{\Delta_i}\right) \log T + \frac{\alpha}{\alpha - 2} \sum_{i:\Delta_i > 0} \Delta_i. \tag{2}$$

$$\mathbb{P}\left\{\widehat{\mu}_i < \mu_i + \sqrt{\frac{2\alpha \log t}{2T_{i,t-1}}} \middle| T_{i,t-1}\right\} \geq 1 - \delta.$$

Let $\delta = \frac{1}{t^\alpha}$ gives the expression for UCB.

Proof Let $\beta := \lceil \frac{2\alpha \log T}{\Delta_i^2} \rceil$. It can be proved by contradiction that in order for arm $i \neq i^*$ to be chosen at time t, at least one of the following three events must occur: 1) $\widehat{\mu}_i \geq \mu_i + \sqrt{\frac{\alpha \log t}{2T_{i,t}^2}}$; 2)$\widehat{\mu}_{i^*} \leq \mu_{i^*} \sqrt{\frac{\alpha \log t}{2T_{i,t}^2}}$; 3) $T_{i,t} \leq \beta$. Therefore, the expected exploration time for arm i can be upper bounded as follows:

$$
\begin{aligned}
\mathbb{E}[T_i] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}[i_t = i]\right] \\
&= \sum_{t=1}^{T} \mathbf{1}[i_t = i, T_i \leq \beta] + \mathbf{1}[i_t = i, T_i \geq \beta] \\
&\leq \beta + \sum_{t=\beta+1}^{T} \mathbb{E}\left[\mathbf{1}[\widehat{\mu}_{i,t} \geq mu_i + \sqrt{\frac{\alpha \log t}{2T_{i,t}^2}}, T_{i,t} \geq \beta] + \mathbf{1}[\widehat{\mu}_{i^*,t} \leq \mu_{i^*} \sqrt{\frac{\alpha \log t}{2_i^2}}, T_{i,t} \geq \beta]\right] \\
&\leq \beta + \sum_{t=\beta+1}^{T} \mathbb{P}\left\{\widehat{\mu}_{i,t} \geq mu_i + \sqrt{\frac{\alpha \log t}{2T_{i,t}^2}}\right\} + \mathbb{P}\left\{\widehat{\mu}_{i^*,t} \leq \mu_{i^*} - \sqrt{\frac{\alpha \log t}{2T_{i,t}^2}}\right\} \\
&= \beta + \sum_{t=\beta+1}^{T} \sum_{t'=\beta+1} t\mathbb{P}\left\{\widehat{\mu}_{i,t} \geq mu_i + \sqrt{\frac{\alpha \log t}{2t'^2}}, T_{i,t} = t'\right\} + \mathbb{P}\left\{\widehat{\mu}_{i^*,t} \leq \mu_{i^*} - \sqrt{\frac{\alpha \log t}{2t'^2}}, T_{i,t} = t'\right\} \\
&\leq \beta + \sum_{t=\beta+1}^{T} t \times \frac{1}{t^\alpha} + t \cdot \frac{1}{t^\alpha} \leq \beta + \int_1^\infty \frac{2}{t^{\alpha-1}} \leq \frac{2\alpha \log T}{\Delta_i^2} + 1 + \frac{2}{\alpha - 2} \leq \frac{2\alpha \log T}{\Delta_i^2} + \frac{\alpha}{\alpha - 2}.
\end{aligned}
$$

Plug Eq. (3) into Eq. (1) leads to Eq. (2). ■

Remark 2 *1. Plug back $\delta = \frac{1}{t^\alpha}, \epsilon = \min_i \frac{1}{\delta_i^2}$, we obtain $\mathbb{E}[T_i] \leq \frac{2\log \frac{1}{\delta}}{\epsilon^2} + \frac{1}{1-\delta^{1/\alpha}} = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ and $\mathbb{E}[T] \leq O(\frac{K}{\epsilon^2} \log \frac{1}{\delta})$. As will be shown later, this is also the optimal lower bound for the stochastic MAB, indicating that UCB is in fact the optimal algorithm.*

2. This proof poses no restriction on the reward distribution of the arms other than boundedness. Further, it can be easily extended to subgaussian reward distributions such as the Gaussian distribution.

3.2 Lower bounds on the performance of optimal multi-arm bandit algorithm

Intuitively, the performance of an (ϵ, δ)-optimal best-arm identifier is limited by the closest arm outside the ϵ ball centered around the best arm. Consider the following scenario: Suppose one arm has probability δ, It is useful to consider a special case when there are two arms with reward distribution as Bernoulli random variables $p_1 = p_0 + \epsilon$. In order to be (ϵ, δ)-optimal, the arm needs to be the optimal hypothesis tester for the following hypotheses:

$$
\begin{aligned}
H_0 : &\quad \mu_0 = p_1, \mu_1 = p_0 \\
H_1 : &\quad \mu_0 = p_0, \mu_1 = p_1.
\end{aligned}
$$

From Chapter 12 of the textbook, we know that the optimal error can be lower-bounded via the Bhattacharyya coefficient: $p^*_{e,T} \geq \frac{(1-\epsilon)^T}{4} = \delta \Rightarrow T = \frac{\log(\frac{1}{4\delta})}{\log \frac{1}{p_0^2+p_0\epsilon}}$. For $K-arm$ problem, the best-arm identification problem is a sequential hypothesis testing problem with at most K trial and therefore $\mathbb{E}[T] \geq \frac{K\log(\frac{1}{4\delta})}{\log \frac{1}{p_0^2+p_0\epsilon}} = O(K\log(\frac{1}{4\delta}))$. With a more careful analysis, it turns out this bound can be made tighter to match the upper bound in the previous section, as shown by the following theorem by Mannor and Tsitsiklis (2004).

Theorem 3 *Fix some $\underline{p} \in (0, \frac{1}{2})$. There exists a positive constant δ_0, c_1 that depends only on $\underline{p}$, such that for every $\epsilon \in (0, \frac{1}{2}), \delta \in (0, \delta_0), p \in [0, \frac{1}{2}]^n$, and for every (ϵ, δ)-optimal policy, we have:*

$$\mathbf{E}_p T \geq c_1 \left\{ \frac{|M(p,\epsilon)-1|^+}{\epsilon^2} + \sum_{l \in N(p,\epsilon)} \frac{1}{(p^*-p_l)^2} \right\} \log \frac{1}{8\delta},$$

where $p* = \max_i p_i$,

$$M(p,\epsilon) = \left\{ l : p_l > p* - \epsilon \text{ and } p_l > \underline{p} \text{ ,and } p_{\geq} \frac{\epsilon}{1+\sqrt{1/2}} \right\}$$
$$N(p,\epsilon) = \left\{ l : p_l \leq p* - \epsilon, \text{ and } p_l > \underline{p}, \text{ and } p_{\geq} \frac{\epsilon}{1+\sqrt{1/2}} \right\}.$$

Proof The proof is outlined as follows: suppose i_T is chosen according to some (ϵ, δ)-optimal policy. Without loss of generality, let $p_1 = p*$. Let $\mathbf{E}_l$ and $\mathbf{P}_l$ be the expectation and probability under hypothesis l respectively. By definition, the policy should make the asymptotically optimal decision among the following $K+1$ hypotheses:

$$H_0 : \mu_1 = p_1, \mu_l = p_l, l \neq 1 \tag{3}$$
$$H_l : \mu_l = p_1 + \epsilon, \mu_i = p_i, i \neq l. \tag{4}$$

Let $B_l = \{i_T = l\}$, then this is the error event when any $H_i, i \neq l$ is true and a correct event when H_l is true. The key idea is to observe that $P_0(B_l^c)$ is large by the optimality of the policy which also makes $P_l(B_l^c) = P_l(\text{error})$ large, leading to contradiction. To that end, the proof then proceeds to lower bound the event $S = A_l \cap B_l^c \cap C$, where $A_l = \{T_l \leq 4t_l^*\}, C_l = \{\max_{1 \leq t \leq 4t_l^*} |\widehat{\mu}_t^l T_l - \mu_l T_1| < \sqrt{t_l^* \log(1/(8\delta))}\}$. Clearly $P_l(S)$ provides a lower bound for $P_l(B_l^c)$. To bound $P_l(S)$, use the standard change-of-measure argument: $P_l(S) = \mathbf{E}_1[\mathbf{1}[w \in A_l \cap B_l^c \cap C]] = \mathbf{E}_0[\frac{L_l(w)}{L_0(w)}\mathbf{1}[w \in A_l \cap B_l^c \cap C]]$, where W is the past observations $\{X_{i_{t'},t'}\}_{t'<t}$ at time t. This can be bounded in the following steps:

1. Lower bound $\frac{L_l(w)}{L_0(w)} \geq C(p_l)$ using the properties of S, namely, the exploration time is no too long and the estimated reward for arm l is sufficiently close to its mean;

2. Lower bound $P_0(S) \geq 1 - P_0(A_l^c) - P_0(B_l) - P_0(C_l^c)$ by lower bound $P_0(A), P_0(B), P_0(C)$ as follows: $P_0(A^c)$ can be upper bounded by Markov inequality; $P_0(B)$ is lower

bounded by the optimality of the policy; $P_0(C^c)$ can be upper bounded by noticing that the deviation of total reward sequence from its mean is Martingale and Kolmogorov inequality can be used to bound the maximum of the sequence at any time in terms of the variance of the reward each time which is $p_l(1-p_l) \leq 1/4$ and the deviation. By properly choosing the constants, we can bound $P_0(S) \geq 1/8$;

3. Combined the two steps, $P_l(S) \geq C(p_l)/8 > \epsilon$, leading to contradiction;

4. Use union bound and that for each arm, $T_l \geq O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$, we have $T \geq O(\frac{K}{\epsilon^2}\log\frac{1}{\delta})$, thus proving the theorem.

■

3.3 Multiclass Adaboost

Adaboost algorithm enjoys nice empirical risk reduction and generalization ability. This is illustrated by the upper bound on generalization loss based on surrogate loss function and Radamacher complexity of a absolute convex hull Hajek and Raginsky (2019).

A more general view of the boosting algorithm proposed by Mukherjee and Schapire (2013) takes several attributes of the original algorithm into accounts:

1. The loss function is entries of a sequence cost matrices $C_t \in \mathbb{R}^{K\times K}, t \in 1,\cdots,T$, whose columns satisfies that, if the true labels are $\{y_t\}_{t=1}^T$, $C_t(y_t) \in \mathcal{C}^{eor}(y)$, where $\mathcal{C}^{eor}(y) = \{\mathbf{c} : c(y) \leq c(l)\}$. Note that as a result the set of cost vectors is convex.

2. The base learners perform better than random guessing, called the *weak-learning condition* (WLC):

$$\sum_t w_t C(i,\widehat{y}_t) \geq \sum_t w_t \langle C(i), u_y^\gamma\rangle, \tag{5}$$

where u_y^γ is called a γ-*over-random* distribution: $[\frac{1-\gamma}{k} + \gamma\mathbf{1}[y=1], \frac{1-\gamma}{k} + \gamma\mathbf{1}[y=2], \cdots, \frac{1-\gamma}{k} + [\gamma\mathbf{1}[y=K]$. The loss $C(i) \in \mathcal{C}_1^{eor}(y) = \{\mathbf{c} : c(y) \leq c(l)\}, \|\mathbf{c}\|_1 = 1\} \in \mathcal{C}^{eor}(y)$. In other word,

Applicable to both binary and multiclass setting, we can view the boosting as a online adversarial game: At each round i, the booster tries to come up with a cost matrix $C^i \in \mathbb{R}^{K\times K}$ to penalize the mistakes the base learner makes while the base learner tries to minimize the loss by finding the best base learner in $\mathcal{H}$. Therefore, the general problem problem of boosting becomes:

$$\max_{C_1\in\mathcal{C}^{eor}} \min_{h_1\in\mathcal{H},\mathbf{tr}(C,U^\gamma-\mathbf{1}_{h_1})\geq 0} \cdots \max_{C_1\in\mathcal{C}^{eor}} \min_{h_N\in\mathcal{H},\mathbf{tr}(C,U^\gamma-\mathbf{1}_{h_N})\geq 0} \frac{1}{T}\sum_{t=1}^T L(\mathbf{s}_N(x_t), y_t)$$

This problem can be hard to optimize, but if we restrict the type of probability distribution to the "edge-over-random" distribution $[\frac{1-\gamma}{k}+\gamma, \frac{1-\gamma}{k}, \cdots, \frac{1-\gamma}{k}]$, the problem can

break down into single steps:

$$\phi_i^y(\mathbf{s}) = \max_{c\in\mathcal{C}^{eor}} \min_{p\in\Delta_\gamma} \{\mathbb{E}_{l\sim p}[\phi_{i-1}^y(\mathbf{s}+\mathbf{e}_l)] : \langle c,p\rangle \le \langle c,u^\gamma\rangle\} \tag{6}$$

$$= \mathbb{E}_{l\sim u_y^\gamma}[\phi_{i-1}^y(\mathbf{s}+\mathbf{e}_l)]. \tag{7}$$

The last equality follows from the minimax principle (the objective is linear in p and C on the convex compact set on p and convex set on C). The function ϕ is called a *potential function* by Mukherjee and Schapire (2013). It is easy to check by induction that if $\phi_0(x)$ is a cost function, so will be $\phi_i(x)$. And the y-th column of the cost matrix is related to the potential function by $C_t^i(y) = [\phi^y(\mathbf{s}_t^{i-1}+\mathbf{e}_1)\cdots\phi^y(\mathbf{s}_t^{i-1}+\mathbf{e}_K)]^\top$

Further, as noted by Jung et al. (2017), this expression is amenable to online setting and a general result in online learning is applicable: For the problem on $(X,\mathcal{P},\mathcal{H},\ell)$, if function class H has finite Littlestone dimension d, with probability $1-\delta$, the regret $R_T((h_t)) = O(\sqrt{Td\ln T}+\sqrt{T})$. The weak learning condition for the base learners can be viewed as a special case with empirical risk $L(h_t) = \sum_{t=1}^T \mathbf{w}_t C_t[y_t, h_t(x_t)]$.

The intuitive interpretation of the potential function can be seen through the following two examples.

Example 1 *Let $\phi_i^y(s) = \mathbb{P}\{\max_{l\neq y} s(l)+N_l \ge s(y)+N_r\}$, where we define N_y to be the number of labels classified to be y by the "edge-over-random" classifier, which draws a label randomly according to the distribution u_y^γ. We have:*

$$\mathbb{P}\left\{\max_{l\neq y} s(l)+N_l \ge s(y)+N_r\right\} = \mathbb{P}\left\{\arg\max_{l\in[K]} \mathit{Multinom}\left(i,u_y^\gamma\right)[l] \neq y\right\}$$

$$= 1 - \sum_{(n_1,\cdots,n_k)} \binom{i}{n_1\cdots n_K} \prod_{i=1}^K u_y^{\gamma n_i}(i).$$

Example 2 *Let $\phi_0^y(s) = \sum_{l\neq y} e^{\alpha(s(l)-s(y))}$, then by induction it can be shown that for $a_k = (1+e^\alpha+e^{-\alpha}-u_y^\gamma(k)-u_y^\gamma(y))$:*

$$\begin{aligned}
\phi_i^y(s) &= \mathbb{E}_{l\sim u_y^\gamma}[\phi_{i-1}^y(\mathbf{s}+\mathbf{e}_l)] \\
&= \sum_{k=1}^K (a_k)^{i-1} \sum_{l\neq y} u_y^\gamma \exp\left(\alpha(s(l)+e_k(l)-s(y)-e_k(y))\right) \\
&= \sum_{l\neq y} \exp(s(l)-s(y)) \sum_{k=1}^K (a_k)^{i-1}(e^\alpha+e^{-\alpha}+\sum_{k\neq y,k\neq l} 1) \\
&= \sum_{l\neq y} (\sum_{k=1}^K (a_k)^i) \exp\left(s(l)-s(y)\right). \qquad (8)
\end{aligned}$$

in this case the loss becomes differentiable and results by Zinkevich on adversarial online learning as mentioned in the textbook Hajek and Raginsky (2019) can be used later to show the exponential decay of the number of mistakes. Also notice that this is a generalization of the binary classification case where $\phi_i^y(s) = e^{\alpha y_i}$ for $y_i \in \{-1,1\}$.

Just as the case in binary case, the loss matrix serves as a surrogate loss that upper bounds the number of mistakes the expert makes, and thus we have the following theorem:

Theorem 4 *Suppose weak learners and an adversary satisfy the online weak learning condition with parameters (δ, γ, S), For any T, N sucht that $\delta = O(\frac{1}{N})$, and any adaptive sequence generated by the adversary, the final loss of the onlineMBBM satisfies the following with probability $1 - N\delta$:*

$$\sum_{t=1}^{T} L^{y_t}(\mathbf{s}_t^N) \leq \phi_N^1(0)T + S\sum_{i=1}^{N} w^{i^*}, \tag{9}$$

where $w^{i^*} := \sup_{t\in[T]} w_t^i = \sup_{t\in[T]} \sum_{k=1}^{K} \phi_i^{y_t}(s_t^{i-1} + \mathbf{e}_k) - \phi_i^{y_t}(s_t^{i-1} + \mathbf{e}_1)$.

Proof By the definition of the potential function:

$$\begin{aligned}
\phi_{N-i+1}^y(s_t^{i-1}) &= \mathbb{E}_{l\sim u_y^\gamma}[\phi_{N-i}^y(s_t^{i-1} + \mathbf{e}_l)] \\
&= \langle C_t^i(y), \mathbf{u}_y^\gamma - \mathbf{e}_{l_t}\rangle + \phi_{N-i}^y(s_t^{i-1} + \mathbf{e}_{l_t}) \\
&= w_t^i\langle D_t^i(y), \mathbf{u}_y^\gamma - \mathbf{e}_{l_t}\rangle + \phi_{N-i}^y(s_t^i) \\
&\leq w_t^i S + \phi_{N-i}^y(s_t^i),
\end{aligned} \tag{10}$$

where the last inequality uses the online weak learning condition. Repeatedly applying Eq. (10) and use the fact that $L^y(s_t^N) = \phi_0^{y_t}(s_t^N)$ leads to the result. ■

3.4 Random drawing model and vote-by-majority algorithm

The random drawing model proposed by Jung et al. (2017) turns out to be a powerful theoretical tools to understand the essence of boosting algorithm. The proofs below all

Theorem 5 *Suppose the same condition as in Thm. (4) and $\gamma < \frac{1}{2}$ the number of mistakes the expert s_t^N make satisfies, with probability $1 - N\delta$:*

$$\sum_{t=1}^{T} \mathbf{1}[y_t \neq \widehat{y}_t] \leq (K-1)e^{-\frac{\gamma^2 N}{2}}T + \tilde{O}(K^{5/2}\sqrt{N}S).$$

Therefore in order to achiever error rate ϵ, it suffices to use $N = \Theta(\frac{1}{\gamma^2}\ln\frac{k}{\epsilon})$.

Proof The proof is outlined as follows: without loss of generality, let $y_t \equiv 1, \forall t \in [T]$. Then the booster makes an error if the event $\mathcal{E} = \{\mathbf{0}) = \{\max_{k\neq 1} s_i(k) \geq s_i(1)\}\}$ happens. The probability of such event can be bounded by the following steps:

1. Use union bound: $\mathbb{P}\{\mathcal{E}\} \leq \sum_{k=2}^{k} \mathbb{P}\{N_k \geq N_1\} = (k-1)$.
2. The event that $N_2 \geq N_1$ can be kept track by a random variable $Y_t^k = 1$ if $\widehat{y}_t = k$ and -1 if $y_t = -1$ and 0 otherwise. Therefore, $\mathbb{P}\left\{Y_i^k = 1\right\} = \frac{1-\gamma}{\gamma} + \gamma, \mathbb{P}\left\{Y_i^k = -1\right\} = \frac{1-\gamma}{\gamma}$ and $\{N_k \geq N_1\} = \{\sum_{j=1}^{N} Y_j^k \leq 0\}$. By Azuma-Hoeffding inequality, we can union bound each of such event for all k to obtain the first term of the right-hand side;

3. To bound the second term, if ϕ is the random drawing potential in Ex.1, we have $\phi_N^1(0) = \mathbb{P}\{\mathcal{E}\}$ and with the help Thm. (4), it suffices to upper bound:

$$w^{i^*} = \sup_{t\in[T]} \sum_{k=1}^{K} \phi_i(s_t^{i-1} + \mathbf{e}_k) - \phi_i(s_t^{i-1} + \mathbf{e}_1) \tag{11}$$

$$= \mathbb{P}\left\{ \max_{k'} s(k') + N_{k'} + e_k(k') \geq s(1) + N_1 \geq s(k') + N_{k'} \right\}, \tag{12}$$

using a similar tracking variable approach, it can be shown that this event is bounded by $C'k\frac{\sqrt{k}}{i}$. The proof uses some of the advanced bounds on the relation between binomial random variables and Gaussian random variables called the Berry-Essen theorem.

■

The random drawing model also provides a lower bound on sample complexity as well as number of mistakes for the booster:

Theorem 6 *For any $\gamma \in (0, \frac{1}{4}), (\delta, \epsilon) \in (0,1)$ and $S \geq \frac{k\ln(\frac{1}{\delta})}{\gamma}$, there exists an adversary with a family of learners satisfying the online learning condition with parameters (ϵ, δ, S), an online boosting algorithm requires at least $\Omega(\frac{1}{k^2\gamma^2}\ln(\frac{1}{\epsilon}))$ learners and a sample complexity of $\Omega(\frac{k}{\epsilon\gamma}S)$.*

Proof Construct a set of edge-over-random base classifiers such that for $t \leq T_0 = \frac{kS}{4\gamma}$, $l_t^i \sim u_{y_t}^0 = [\frac{1}{k}, \cdots, \frac{1}{k}]$ and $l_t^i \sim u_{y_t}^{2\gamma}$ otherwise. It turns out that the base learners l_t^i in this case are γ-over-random. For $T \leq T_0$, by Azuma-Hoeffding inequality:

$$\sum_{t=1}^{T} w_t \mathbf{C}_t(y_t, \widehat{y}_t) \leq \sum_{t+1}^{T} w_t \frac{1}{k} + \sqrt{\|w\|_2^2 \ln\frac{1}{\delta}}$$

$$\leq \sum_{t=1}^{T} w_t \frac{1}{k} + \frac{4\gamma\|w\|_1}{k} + \frac{k\ln\frac{1}{\delta}}{4\gamma} \leq \|w\|\frac{1-\gamma}{k} + 2\frac{\gamma}{k}T_0 + S = \|w\|\frac{1-\gamma}{k} + S.$$

Similarly, the inequality holds for $T > T_0$. By Thm. (4) the mistakes made by the set of classifiers go to zero asymptotically. Therefore, $T_0 = O(\frac{kS}{4\gamma})$ is a lower bound for time complexity. Next, to show the lower bound for the number of learners, again use the random drawing model on $\phi_0(s^N)$ and a fact about Binomial distribution called Slud's inequality:

$$\mathbb{P}\{\text{error}\} \geq \mathbb{P}\{N_2 > N_1\} = \mathbb{P}\left\{ \sum_{j=1}^{N} Y_j < 0 \right\}$$

$$\geq \mathbb{P}\left\{ \text{Binom}(m, \frac{p_1}{p_1 + p_{-1}} > \frac{m}{2}) \right\} \geq \Omega(\exp(-4mk^2\gamma^2)) \geq \Omega(\exp(-4Nk^2\gamma^2)).$$

■

3.5 AdaBandit Algorithm

One way MAB can be used in the Adaboost algorithm is in the feature selection step. The same reasoning in the standard stochastic bandit identification setting works with almost no modification. However, since the bandit is applied in each full pass of the dataset, the algorithm is not online and still has large rooms for improvement.

Another way MAB can be applied is at the selection of experts in Adaboost. Suppose we have a sequence of experts $\{s_i\}_{i=1}^N$ formed by the base learners. If the Adaboost.OLM is used, we need to choose the best expert. This can be seen as an best arm identification problem and if Hedge algorithm is used, whose analysis will be left out due to page constraints, the number of errors made by the chosen expert will be no more than $2\min iM_i + 2\log N + \ O(\sqrt{T})$ than the best expert. Notice that in the previous section, we assume that the base classifiers all have the same edge γ. Jung et al. (2017) introduce a suboptimal algorithm called Adaboost.OLM. In this algorithm, $L^r(s) = \sum_{l\neq 1} \log(1 + \exp(s(l) - s(r)))$ and in general $L^r(s)$ can be any differentiable function that is monotonically increasing in $s(l), l \neq r$ and decreasing in $s(r)$. As in the binary classification case, the loss can be minimized one base learner at a time using gradient descent:

$$\mathbf{s}_t^i = \mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{e}_{l_t^i}$$
$$\alpha_{t+1}^i = \Pi(\alpha_t^i - \eta_t \frac{\partial L^r(\mathbf{s}_t^i)}{\partial \alpha}).$$

entries of the cost matrix are the gradients of $L^r(s)$ with respect to s. To select the optimal expert, the *Hedge algorithm* Littlestone and Warmuth (1989) is used, which randomly choose an expert with probability $v_{t+1}^i = \frac{\exp(-\eta \sum_t \mathbf{1}[y_t \neq \widehat{y}_t^i, i_t = i])}{\sum_{j=1}^N \exp(-\eta \sum_{t=1}^T \mathbf{1}[y_t \neq \widehat{y}_t^i, i_t = j]]}$.

Theorem 7 *For any T and N, with probability $1-\delta$, the number of mistakes made by Adaboost.OLM satisfies the following inequality:*

$$\sum_{t=1}^T \mathbf{1}[y_t \neq \widehat{y}_t] \leq \frac{8(k-1)}{\sum_{i=1}^N \gamma_i^2} T + \tilde{O}(\frac{kN^2}{\sum_{i=1}^N \gamma_i^2}).$$

Proof Here is a sketch of the proof:

1. First it can be verified that, the derivative of the logistic loss with respect to *alpha* is:

$$\frac{\partial L^r(s_t^{i-1})}{\partial \alpha} := \begin{cases} \frac{1}{1+\exp(s_t^{i-1}(r) - s_t^{i-1}(j))} & \text{, if } l \neq r \\ -\sum_{j\neq r} \frac{1}{1+\exp(s_t^{i-1}(r) - s_t^{i-1}(j))} & \text{, if } l = r. \end{cases}$$

   Notice that the gradient of the loss is a cost matrix from the property of the loss and in this case, the magnitude of the loss is bounded by $(k-1)$, therefore the loss is $(k-1)$-Lipschitz continuous.

2. Treat the algorithm as an online gradient descent problem on α and apply a standard analysis in online convex optimization by Zinkevich Zinkevich (2003): let $\Delta_i := L^{y_t}(s_t^i) - L^{y_t}(s_t^{i-1})$, we have

$$\sum_t \Delta_i \leq 2 \min_{\alpha \in [-2,2]} \sum_t L^{y_t}(s_t^i) - L^{y_t}(s_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) + 4\sqrt{2}(k-1)\sqrt{T}. \tag{13}$$

3. Next, notice that the first term on the right-hand side is upper bounded by an expression of its derivative, which seems to be a special property of Adaboost algorithm: the exponential loss in the binary case also enjoys relation between the loss and its derivative:

$$\min_{\alpha \in [-2,2]} \sum_t \sum_i L^{y_t}(s_t^i) - L^{y_t}(s_t^{i-1}) \leq \sum_t \sum_i (\exp(\alpha) - 1) \sum_{t: l_t^i \neq y_t} C_t^{i-1}(y_t, l_t^i) + (-\exp(-\alpha) + 1) \sum_{t: l_t = y_t} C_t^{i-1}(y_t, l_t^i) \leq -\frac{\gamma_i^2}{2} w^i$$

4. Observe that the number of mistakes made by base learner i: $M_i := \sum_{t=1}^T \mathbf{1}[\widehat{y}_t^i \neq y_t]$ is upper bounded by $2\sum_{t=1}^T C^i(y_t, \widehat{y}_t^i)$: if expert $i-1$ makes a mistake, at least one term in $-C^i[r, r]$ is no less than $1/2$. Therefore $-w^i \leq -\frac{M_i}{2}$. Combined the steps and moved M_i to the left-hand side and take the min and apply the bound on Hedge algorithm leads to the result.

■

Notice that similar bound holds for exponential loss and hinge loss as well since they also have close relations to their derivatives Jung et al. (2017).

4. Conclusion and Future works

This project has covered a set of problems related to bandit-aided boosting. Due to time constraints, I was not able to perform numerical experiments to test the theory presented in the papers. In the future, I would like to implement a few boosting algorithm and see the gap between theory and practice. Further, I would like to derive tighter lower bound for Adaboost.OLM and bandit.OLM. As mentioned above, an alternative way to formulate bandit-aided boosting will be to use stochastic bandits for feature selection and I would like to understand the theoretical guarantee of it in the online boosting setting.

Data: $(x_t, y_t), t = 1, \cdots, T$; a set of weak classifiers $\{h_i\}_{i=1}^N$
Result: best expert $\mathbf{s}^N$
Initialize the sample weights $w_t = \frac{1}{T}$
for $t := 1 : T$ **do**
  Receive data $\mathbf{x}_t$
  for $i := 1 : N$ **do**
    Compute the cost matrix
    $C_t^i = [\mathbb{E}_{l\sim u^\gamma}\phi_{N-i}^1(s_t^{i-1} + \mathbf{e}_l), \cdots, \mathbb{E}_{l\sim u^\gamma}\phi_{N-i}^K(s_t^{i-1} + \mathbf{e}_l)]$ according to Eq. (**??**)
    $k_m^* = k_{m,T}$ and normalize it to D_t^i according to Eq. (**??**)
  end
  Weak learner i predicts the label l_t^i based on D_t^i
  Booster computes the learner weight α_t^i and update the votes $s_t^i = s_t^{i-1} + \mathbf{e}_{l_t^i}$
  Choose the expert i^* according to distribution $\mathbf{v}_t$ and makes a prediction $\widehat{y}_t$
  Receive true label $\mathbf{y}_t$
  for $i := 1 : N$ **do**
    Set $\mathbf{w}^i[t] := \sum_l [\phi_{N-i}^{y_t}(\mathbf{s}_t^{i-1} + \mathbf{e}_l) - \phi_{N-i}^{y_t}(s_t^{i-1} + \mathbf{e}_{y_t})]$
    Passing training example with weight $(\mathbf{x}_t, y_t, \mathbf{w}^i[t])$ to the weak learner i
    Update expert distribution $\mathbf{v}_t$
  end
end

Algorithm 1: General pseudocode for bandit boosting algorithm

References

Ròbert B. Fekete and Balázs Kegl. Bandit-aided boosting. 2009. URL `http://opt.kyb.tuebingen.mpg.de/papes/OPT2009-BusaFekete.pdf`.

Bruce Hajek and Maxim Raginsky. ECE 543: Statistical learning theory, 2019. URL `https://courses.engr.illinois.edu/ece543/sp2019/SLT.pdf`.

Y. H. Jung, J. Goetz, and A. Tewari. Online multiclass boosting. 2017. URL `https://arxiv.org/pdf/1702.07305.pdf`.

Dimitrios Katselis. Ece 586 mdps and reinforcement learning lecture 8: Multi-armed bandits. URL `http://katselis.web.engr.illinois.edu/ECE586/Lecture8.pdf`.

N. Littlestone and Manfred K. Warmuth. The weighted majority algorithm, 1989.

Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. 2004. URL `http://www.jmlr.org/papers/volume5/mannor04b/mannor04b.pdf`.

I. Mukherjee and R.E. Schapire. A theory of multiclass boosting. In *Journal of Machine Learning Research*, volume 14, pages 437–497, Feb 2013. URL `http://rob.schapire.net/papers/multiboost-journal.pdf`.

Robert E. Schapire and Yoram Singer.

P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade, 2002. URL http://papers.nips.cc/paper/2091-fast-and-robust-classification-using-asymmetric-adaboost-and-a-detector-cascade.pdf.

Daniel Zhang, Y. H. Jung, and A. Tewari. Online multiclass boosting with bandit feedback, 2019. URL https://arxiv.org/pdf/1810.05290.pdf.

Yuan Zhou, Xi Chen, and Jian Li. Optimal pac multiple arm identification with applications to crowdsourcing, 2014. URL http://proceedings.mlr.press/v32/zhoub14.html.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent, 2003.