Bandit Multiclass Classification

1 Problem Statement

A multiclass linear classifier is defined by a matrix $W \in \mathbb{R}^{K\times d}$, where K is the number of classes ($K \geq 3$) and d is the dimension of the feature vector. A set of samples $\{(x_t, y_t)\}_{t=1}^T$ where $x_t \in \mathbb{R}^d$ and $y_t \in [K]$ is called *linearly separable* if there exists a W^* such that $y_t = \operatorname{argmax}_{y\in[K]}(W^* x_t)_y$ for all $t \in [T]$. (Note that $(W^* x_t)_y$ can also be written as $\mathbf{e}_y^\top W^* x_t$)

A set of samples $\{(x_t, y_t)\}_{t=1}^T$ is called *linearly separable with a margin* γ if there exists a W^* with $\|\mathbf{e}_i^\top W^*\|_2 \leq 1$ for all i and $(W^* x_t)_{y_t} \geq \max_{y\neq y_t}(W^* x_t)_y + \gamma$ for all t. For simplicity, we define the set where W^* lies in as $\mathcal{W} = \{W \in \mathbb{R}^{K\times d} : \|\mathbf{e}_i^\top W^*\|_2 \leq 1\}$.

Consider the following *online multiclass classification* problem: at time t, the environment first reveals $x_t \in \mathbb{R}^d$ (and we assume $\|x_t\|_2 \leq 1$), then the learner predicts some label $\tilde{y}_t \in [K]$; finally the environment reveals the true label $y_t \in [K]$. It is known that if the samples are linearly separable, then the learner will only make constant mistakes. It is achievable with the following simple perceptron algorithm:

Algorithm 1: Perceptron

$W_1 = \mathbf{0}$
for $t = 1$ *to* T **do**
  Predict $\tilde{y}_t = \operatorname{argmax}_y (W_t x_t)_y$
  Update $W_{t+1} \leftarrow W_t + (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) x_t^\top$.

The analysis is simple. On one hand we have (note $\langle A, B\rangle := \mathrm{Tr}(A^\top B)$):

$$\begin{aligned}\langle W_t, W^*\rangle &= \langle W_{t-1}, W^*\rangle + \langle (\mathbf{e}_{y_t} - \mathbf{e}_{\tilde{y}_t}) x_t^\top, W^*\rangle \\ &= \langle W_{t-1}, W^*\rangle + (\mathbf{e}_{y_t}^\top W^* x_t - \mathbf{e}_{\tilde{y}_t}^\top W^* x_t) \\ &\geq \langle W_{t-1}, W^*\rangle + \gamma \mathbf{1}[\tilde{y}_t \neq y_t].\end{aligned}$$

By induction, we get $\langle W_{T+1}, W^*\rangle \geq \gamma \sum_{t=1}^T \mathbf{1}[\tilde{y}_t \neq y_t]$.

On the other hand,

$$\langle W_{T+1}, W^*\rangle \leq \|W_{T+1}\|_F \|W^*\|_F \leq \sqrt{2K \sum_{t=1}^T \mathbf{1}[\tilde{y}_t \neq y_t] \|x_t\|_2^2} \times 1 \leq \sqrt{2K \sum_{t=1}^T \mathbf{1}[\tilde{y}_t \neq y_t]}$$

Combining the above two inequality, we get $\sum_{t=1}^T \mathbf{1}[\tilde{y}_t \neq y_t] \leq \frac{4K}{\gamma^2}$.

The *bandit multiclass classification* problem proceeds in a very similar way. The only difference is that the learner only knows whether she predicts **correctly or not**, rather than gets the true label (i.e., in each round, the feedback is only $\mathbf{1}[\tilde{y}_t = y_t]$, rather than y_t). Surprisingly, with this extremely

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

limited feedback, the learner can still make only constant mistakes. It can be achieved with the following variant of *halving algorithm*:

Algorithm 2: Halving

Discretize the space of W (i.e., the $\mathcal{W}$ as defined in the beginning) with balls of radius $\frac{1}{2\gamma}$. Let the set of discretization points be $\mathcal{S}$. Then this is saying that for all $W \in \mathcal{W}$, there is always a $W' \in \mathcal{S}$ such that $\|W - W'\|_F \leq \frac{1}{2\gamma}$.
Let $\mathcal{S}_1 = \mathcal{S}$
for $t = 1$ *to* T **do**
    Let $\mathcal{S}_t(y) = \{W \in \mathcal{S}_t : (Wx_t)_y \geq (Wx_t)_i \ \forall i \in [K]\}$ (i.e., the set of W's in $\mathcal{S}_t$ that predict y as the label of x_t)
    Let $\tilde{y}_t = \text{argmax}_y \, |\mathcal{S}_t(y)|$ (i.e., pick the majority vote)
    if $\tilde{y}_t \neq y_t$ **then**
        $\mathcal{S}_{t+1} = \mathcal{S}_t \backslash \mathcal{S}_t(\tilde{y}_t)$ (i.e., eliminate the W's that are inconsistent with the outcomes)
    else
        $\mathcal{S}_{t+1} = \mathcal{S}_t$

Analysis. The majority in $\mathcal{S}_t$ always has cardinality no less than $\frac{1}{K}|\mathcal{S}_t|$. So every time the learner predicts incorrectly, the size of $|\mathcal{S}_t|$ shrinks by an order of $(1 - \frac{1}{K})$. By our margin assumption, there is a $W' \in \mathcal{S}$ which incurs no error in all samples. Therefore, $|\mathcal{S}_t| \geq 1$ always holds. This means the number of errors is bounded by the order of $\frac{\log(|\mathcal{S}|)}{-\log(1-\frac{1}{K})} \leq \mathcal{O}\left(K^2 d \log \frac{1}{\gamma}\right)$.

The main issue of this halving algorithm is that it is **inefficient** in the sense that $|\mathcal{S}|$ is in the order of $\frac{1}{\gamma^{Kd}}$.

Hence, we want to answer the following question: for bandit multiclass classification, can we have an efficient algorithm (time complexity polynomial in $K, d, \frac{1}{\gamma}, T$) that guarantees constant mistake bound (polynomial in $K, d, \frac{1}{\gamma}$) in the γ-separable case? (An unsolved open problem in [4])

2 Naive Approaches

- **Exhausting the feature space**: discretize the feature space into $\left(\frac{1}{\gamma}\right)^{\mathcal{O}(d)}$ blocks. The margin assumption guarantees that two points in a block would have the same label (so we can give every block a label). Each time a new point x_t comes, see whether the learner already knows the correct label for that block. If yes, predict that label; if not, randomly choose a label that has not been chosen for that block.
  $\Rightarrow$ error bound: $K\left(\frac{1}{\gamma}\right)^{\mathcal{O}(d)}$.
- **Exhausting the hypothesis space**: Discretize the hypothesis space into $\left(\frac{1}{\gamma}\right)^{Kd}$ discretization points (like in the halving algorithm). For each of them, make predictions based on it until it makes an mistake.
  $\Rightarrow$ error bound: $\left(\frac{1}{\gamma}\right)^{Kd}$.

3 Difficulties

Efficient classification algorithms often optimize a convex surrogate loss rather than a 0-1 loss. Examples include:

- Hinge loss: $\ell_t(W) = [1 - (Wx_t)_{y_t} + \max_{y \neq y_t}(Wx_t)_y]_+$.
- Logistic loss: $\ell_t(W) = -\ln \frac{\exp((Wx_t)_{y_t})}{\sum_y \exp((Wx_t)_y)}$
- A family of loss that interpolates hinge loss and squared hinge loss (see [1]'s Eq.(3))

In full-information setting, one can directly use online convex optimization techniques to deal with the above loss functions (learning over the space of W).

Many works for bandit classification reuse this kind of convex optimization schemes, together with explicit exploration and inverse propensity weighting [4, 3, 1, 2]. However, because it is hard to estimate these losses when the true label y_t is not known, some of these algorithms [1, 2] simply do not update when the learner makes a mistake ($\tilde{y}_t \neq y_t$). We give this kind of algorithm an error lower bound $\Omega\left(\left(\frac{1}{\gamma}\right)^{(d-1)/2}\right)$ in Section 7.

It indeed looks hard to design a convex loss for W's when the learner makes a mistake: when $\tilde{y}_t \neq y_t$, the set of W's that we want to penalize (i.e., to assign larger loss) is $\{W : (Wx_t)_{\tilde{y}_t} \geq (Wx_t)_y, \forall y\}$, which is a convex cone in the space of W. It is impossible for a convex function to be large only in a convex subset (the case $K = 2$, on the other hand, does not have this issue). Can we argue that if the learner is restricted to use convex losses in the space of W, she will have to suffer exponential (in K) errors?

4 One-versus-all Perceptron

Our assumption of linearly separable with a margin is

$$(W^* x_t)_{y_t} \geq (W^* x_t)_y + \gamma, \ \forall y \neq y_t. \tag{1}$$

As discussed in Section 3, it seems hard to make update when $\tilde{y}_t \neq y_t$ and achieve a constant and polynomial error bound.

For a moment, in this subsection we make the following stronger margin assumption (which we call *one-versus-all separable* with a margin):

$$\begin{cases} (W^* x_t)_{y_t} \geq \gamma/2 \\ (W^* x_t)_y \leq -\gamma/2, \forall y \neq y_t. \end{cases} \tag{2}$$

Clearly, one-versus-all separability implies linearly separability, but not the other way around. With one-versus-all separability assumption, we can view the problem as K parallel binary classification problem. The following algorithm achieves constant error bound:

Algorithm 3: One-versus-all Perceptron

1 Initialize $w_t^{(1)} = \ldots = w_t^{(K)} = \mathbf{0} \in \mathbb{R}^d$
2 **for** $t = 1$ *to* T **do**
3 **if** $\exists y$ *such that* $w_t^{(y)\top} x_t \geq 0$ **then**
4 Assign $\tilde{y}_t$ to any y with $w_t^{(y)\top} x_t \geq 0$
5 Predict $\tilde{y}_t$
6 **if** $\tilde{y}_t \neq y_t$ **then** update $w_t^{(\tilde{y}_t)} \leftarrow w_t^{(\tilde{y}_t)} - x_t$;
7 **else**
8 Pick $\tilde{y}_t$ randomly from unif$[K]$
9 Predict $\tilde{y}_t$
10 **if** $\tilde{y}_t = y_t$ **then** update $w_t^{(\tilde{y}_t)} \leftarrow w_t^{(\tilde{y}_t)} + x_t$;

Analysis. Note that when the algorithm enters Line 6 or Line 10, the binary classifier $w_t^{(\tilde{y}_t)}$ is making an error. Let the number of times the algorithm enters Line 6 and Line 10 be M and N respectively. By the error bound of binary perceptron $\mathcal{O}\left(\frac{1}{\gamma^2}\right)$, we have that $M + N = \mathcal{O}\left(\frac{K}{\gamma^2}\right)$. Then note that the number of times the algorithm makes a mistake (i.e. $\tilde{y}_t \neq y_t$) is upper bounded by M plus how many times the algorithm explores in Line 8; and when the algorithm explores, with probability $\frac{1}{K}$ it enters Line 10. Therefore, the number of mistakes is bounded (in expectation) by $\mathbb{E}[M + KN] = \mathcal{O}\left(\frac{K^2}{\gamma^2}\right)$.

5 One-versus-all Kernel Perceptron

With the polynomial kernel defined in [5], we can transform the weaker assumption (1) in the original feature space to the stronger one (2) in a transformed feature space. Indeed, in the original feature space, class i corresponds to an intersection of $K-1$ halfspaces: $\{x : (w_i^* - w_j^*)^\top x \geq 0, \forall j \neq i\}$, which is the subject of [5] (here, $w_i^{*\top}$ equals to $\mathbf{e}_i^\top W^*$ defined above). Directly using their kernel construction, we can have margin $\left(\frac{\gamma}{d}\right)^{\Omega(K \log K \log(1/\gamma))}$ in the transformed feature space. Running kernelized version of Algorithm 3, we can get $K^2 \left(\frac{d}{\gamma}\right)^{\mathcal{O}(K \log K \log(1/\gamma))}$ error bound directly.

5.1 Refinement for [5]'s Fact 1

Lemma 1. *For $i = 1, \ldots, \ell$ let $q^{(i)}(x) = \sum_S c_S^{(i)} x_S$ be a polynomial over $x_1, \ldots, x_d$ with $\|q^{(i)}\|^2 \leq M_i$. Then (1) if $q^{(1)} \ldots q^{(\ell)}$ has degree at most $\deg$, we have $\|q^{(1)} \ldots q^{(\ell)}\|^2 \leq (\deg)^{O(\deg)} \prod_i M_i$, and (2) we have $\|q^{(1)} + \ldots + q^{(\ell)}\|^2 \leq \ell(M_1 + \ldots + M_\ell)$.*

Proof. For the first bound, we bound the ratio between the following two values: $\|q^{(1)} \ldots q^{(\ell)}\|^2$ and $\|q^{(1)}\|^2 \ldots \|q^{(\ell)}\|^2$.

$$n_1 = \prod_{i=1}^{\ell} \|q^{(i)}\|^2 = \prod_{i=1}^{\ell} \left(\sum_{S_i} \left(c_{S_i}^{(i)} \right)^2 \right) = \sum_{(S_1,\ldots,S_\ell)} \left(\prod_{i=1}^{\ell} \left(c_{S_i}^{(i)} \right)^2 \right) = \sum_{(S_1,\ldots,S_\ell)} \left(\prod_{i=1}^{\ell} c_{S_i}^{(i)} \right)^2$$
$$= \sum_S \sum_{\substack{(S_1,\ldots,S_\ell): \\ S_1 \times \ldots S_\ell = S}} \left(\prod_{i=1}^{\ell} c_{S_i}^{(i)} \right)^2 \tag{3}$$

$$n_2 = \|q^{(1)} \ldots q^{(\ell)}\|^2 = \sum_S c_S^2 = \sum_S \left(\sum_{\substack{(S_1,\ldots,S_\ell): \\ S_1 \times \ldots \times S_\ell = S}} \left(c_{S_1}^{(1)} \ldots c_{S_\ell}^{(\ell)} \right) \right)^2 = \sum_S \left(\sum_{\substack{(S_1,\ldots,S_\ell): \\ S_1 \times \ldots \times S_\ell = S}} \prod_{i=1}^{\ell} c_{S_i}^{(i)} \right)^2 \tag{4}$$

Let M be an upper bound of the number of terms involved in the summation $\sum_{\substack{(S_1,\ldots,S_\ell): \\ S_1 \times \ldots \times S_\ell = S}}$, then by Cauchy-Schwarz's inequality we have $n_2 \leq M n_1$.

M counts the number of different $(S_1, \ldots, S_\ell)$'s with $S_1 \times \ldots \times S_\ell = S$. Since S has degree at most $\deg$, we can bound M by $\ell^{\deg}$.

The second bound can be obtained by applying Cauchy-Schwarz once.

□

5.2 Refinement for [5]'s Theorem 10

Lemma 2. *Under the same condition as in [5]'s Theorem 10, the PTF has margin on X is at least $(1/t)^{O(r \log r + r \log\log t)}$.*

Proof. We simply follow the construction in the original paper. $\|1 - w^i \cdot x\|^2 \leq 4$ can imply $\|(1 - w^i \cdot x)^j\|^2 \leq j^{O(j)} 4^j \leq j^{O(j)}$. Then $\|a_j (1 - w^i \cdot x)^j\|^2 \leq 2^{2r} r^{O(r)} = r^{O(r)}$ follows. Thus $\|T_r(1 - w^i \cdot x)\|^2 = \|\sum_{j=0}^r a_j (1 - w^i \cdot x)^j\|^2 \leq (r+1)^2 r^{O(r)} = r^{O(r)}$. Then, $\left(P(w^i \cdot x)\right)^{\log 2t} \leq (\log t)^{O(r \log t)} \times r^{O(r \log t)} \leq (r \log t)^{O(r \log t)}$. Finally, $\|p\|^2 \leq (t+1)^2 (r \log t)^{O(r \log t)} \leq (t+1)^2 \left(t^{O(r \log r + r \log\log t)}\right) = t^{O(r \log r + r \log\log t)}$. □

5.3 Refinement for [5]'s Theorem 7

Lemma 3. *Under the same condition as in [5]'s Theorem 7, the PTF has margin on X is at least* $\left(\frac{\rho}{t\log t\log\frac{1}{\rho}}\right)^{O(t\log t\log 1/\rho)}$.

Proof. Using the same construction, we bound $\|p\|$. First, we have $\|\frac{2w^i\cdot x}{\rho}\|^2 \le \frac{4}{\rho^2}$ and $\|\left(\frac{2w^i\cdot x}{\rho}\right)^j\|^2 \le j^j\left(\frac{4}{\rho^2}\right)^j = \left(\frac{4j}{\rho^2}\right)^j$. $a(x), b(x)$ are polynomials of degree $O(\log t\log\frac{1}{\rho})$ with coefficients of magnitude $\left(\frac{1}{\rho}\right)^{O(\log t\log 1/\rho)}$. Thus

$$\begin{aligned}\|a(2w^i\cdot x/\rho)\|^2 &\le \left(\frac{\log t\log\frac{1}{\rho}}{\rho}\right)^{O(\log t\log 1/\rho)} \times \left(\frac{1}{\rho}\right)^{O(\log t\log 1/\rho)} \\ &\le \left(\frac{\log t\log\frac{1}{\rho}}{\rho}\right)^{O(\log t\log 1/\rho)}.\end{aligned}$$

Same holds for $\|b(2w^i\cdot x/\rho)\|^2$. Finally we have $\|B(x)\|^2 \le (t)^{O(t\log t\log 1/\rho)} \times \left(\frac{\log t\log\frac{1}{\rho}}{\rho}\right)^{O(t\log t\log 1/\rho)} = \left(\frac{t\log t\log\frac{1}{\rho}}{\rho}\right)^{O(t\log t\log 1/\rho)}$ □

6 Regret Lower Bound for Explore-then-Exploit Algorithms [TODO]

7 A Regret Lower Bound for A Certain Type of Algorithms

In this section, we try to construct an error lower bound for a certain type of algorithms. This type of algorithms does not make update when it makes a wrong prediction. For simplicity, we only consider binary classification. More formally, the algorithms we consider satisfy the following assumption.

Assumption 1 (Algorithm)**.** *Let $p_t(x)$ be the algorithm's probability of predicting class 1 (recall we consider binary classification) at round t if it receives the feature vector $x \in \mathcal{X} \subset \mathbb{R}^d$. We assume $p_t(\cdot)$ is totally determined by all previous **correct** examples. In other words, $p_t(\cdot)$ is determined by the tuple $((x_{\tau_1}, y_{\tau_1}), \ldots, (x_{\tau_N}, y_{\tau_N}))$ where $1 \le \tau_1 < \tau_2 \ldots < \tau_N < t$ are the rounds that the learner makes correct prediction.*

Assumption 2 (linearly separable with a margin)**.** *We assume the samples are linearly separable with margin γ (i.e., any two points with different labels have distance no less than γ).*

Definition 4 (Free space)**.** *The free space at time t is the set of points whose label is still under-determined given $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$. For example, the γ-ball centered around any already presented point is excluded from the free space. Denote the free space at time t by FS_t.*

The free space's definition simply means that at time t, the adversary can pick any point x_t in FS_t and assign the label y_t to either 1 or 2 without violating the linearly separable and the γ-margin assumption.

Below we present the Adversary's strategy of constructing (x_t, y_t).

Algorithm 4: Adversary's strategy

Pick x_1 randomly from $\mathcal{X}$, and let $y_1 = 1$.
for $t = 2, \ldots, T$ **do**
  if $\tilde{y}_{t-1} \neq y_{t-1}$ **then**
    Let $(x_t, y_t) = (x_{t-1}, y_{t-1})$
  else if *FS_t is not empty* **then**
    Pick $x_t \in \text{FS}_t$. Because of this x_t, the free space's volume is reduced. We denote the reduction amount by $V_t = v(\text{FS}_{t+1}) - v(\text{FS}_t) \leq V$. (i.e., V is a global upper bound of V_t)
    If $p_t(x_t) \geq 1 - \max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}$, then label $y_t = 2$; otherwise, label $y_t = 1$.
  else
    Randomly assign (x_t, y_t) with some value that does not violate the assumption.

Definition 5 (history)**.** *Let $\mathcal{H}_t$ be the history before time t: $\mathcal{H}_t = \{(x_s, y_s, \tilde{y}_s)\}_{s=1}^{t-1}$. We use $\mathbb{E}_t[\cdot]$ to denote $\mathbb{E}[\cdot|\mathcal{H}_t]$.*

Lemma 6. *If $\exists t$ such $p_t(x_t) \geq 1 - \max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}$, then $\mathbb{E}_t\left[\sum_{s=t}^{T} \mathbf{1}[\tilde{y}_s \neq y_s]\right] \geq \Omega\left(\min\left\{\frac{1}{\sqrt{V}}, \sqrt{T}\right\}\right)$.*

Proof. By of the condition and the adversary strategy, we have $y_t = 2$. Therefore, the learner will predict the true label with probability $\leq \max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}$. And note that if the learner predicts incorrectly at time t, then at time $t+1$ the feature vector remains the same $(x_{t+1} = x_t)$, and the learner's probability of prediction also remains the same $(p_{t+1}(\cdot) = p_t(\cdot))$. Therefore, the expected number of mistakes before the first correct guess is (roughly) larger than $\frac{1}{\max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}} = \min\left\{\frac{1}{\sqrt{V}}, \sqrt{T}\right\}$. □

Lemma 7. *If $\forall s$, $p_s(x_s) \leq 1 - \max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}$, then $\sum_{s=1}^{T} \mathbb{E}_s[\mathbf{1}[\tilde{y}_s \neq y_s]] = \Omega\left(\min\left\{\sqrt{T}, \frac{1}{\sqrt{V}}\right\}\right)$.*

Proof. By the condition and the adversary strategy, we know that the probability of error is larger than $\max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\}$ at all time t *before the free space is used up*. Since each time the free space only reduces by V, in the first $\frac{1}{V}$ rounds (assume the total volume is 1), the free space is still all available. Therefore,

$$\sum_{s=1}^{T} \mathbb{E}_s[\mathbf{1}[\tilde{y}_s \neq y_s]] \geq \min\left\{T, \frac{1}{V}\right\} \times \max\left\{\sqrt{V}, \frac{1}{\sqrt{T}}\right\} = \min\left\{\sqrt{T}, \frac{1}{\sqrt{V}}\right\}.$$

□

Discussion. There is a construction such that $1/V$ can be of order $\Omega\left(\left(\frac{1}{\gamma}\right)^{(d-1)/2}\right)$.

8 How hard it is to use the feedback only from wrong guesses?

The halving algorithm can actually run if we can do "uniform sampling" over the version space. But it is even unknown whether we can efficiently pick a model from the version space. The problem is that we get a lot of feedback in the form of "feature x_t does not belong to class $\tilde{y}_t$", which we don't know how to use.

The following is just an attempt (not successful but might be interesting...) to say that it might be not easy to figure out a model if the learner is only presented with this kind of "error message".

The problem is formulated as follows. Given N points in a row, each one with a class $c_i \in [K]$, $\forall i \in [N]$. We call these N points *separable* if the following statement holds:

$$\text{If } c_i = c_j \text{ for some } i \leq j, \text{ then } c_i = c_{i+1} = \cdots = c_j.$$

For example, if $N = 5, K = 3$, then $(c_1, c_2, c_3, c_4, c_5) = (3, 3, 1, 1, 1)$ is separable, but $(c_1, c_2, c_3, c_4, c_5) = (2, 1, 2, 2, 2)$ is not.

Now you have N conditions, in which the i-th condition only says something like "$c_i \neq k$" for some k.
(1) Can you efficiently decide whether there exists an assignment of $(c_1, \ldots, c_N)$ such that these N points are separable and satisfy all the conditions?
(2) If it is guaranteed that there are separable solutions, can you efficiently find one of them?
(Efficient: the complexity is polynomial in N and K)

Example 1.
$N = 5, K = 3$:
$c_1 \neq 1$
$c_2 \neq 2$
$c_3 \neq 3$
$c_4 \neq 1$
$c_5 \neq 2$
$\Rightarrow (c_1, c_2, c_3, c_4, c_5) = (2, 1, 1, 3, 3)$ or $(3, 3, 2, 2, 1)$ are separable solutions.

Example 2.
$N = 7, K = 3$:
$c_1 \neq 1$
$c_2 \neq 2$
$c_3 \neq 3$
$c_4 \neq 1$
$c_5 \neq 2$
$c_6 \neq 3$
$c_7 \neq 1$
$\Rightarrow$ There is no separable solution.

It turns out this specific 1-dimensional problem is equivalent to identify a **missing permutation** of $[K]$ as a subsequence in the given sequence. In Example 1, the existing permutations are $(1, 2, 3), (1, 3, 2), (2, 3, 1), (3, 1, 2)$, the missing ones are $(2, 1, 3)$ and $(3, 2, 1)$. So the solutions can be $(2, 1, 3)$ or $(3, 2, 1)$ (with some repetition). In Example 2, all permutations are as subsequences, so there is no solution.

We can prove this equivalence considering two directions:
(1) If there is a solution with the class labels following a permutation, then that permutation cannot be a subsequence of the given sequence.
(2) If there is a missing permutation in the given sequence, then there is a class assignment that follows this permutation.
They should be straightforward by trying some examples.

9 Biased Halving: Trading Error with Complexity

Algorithm 5: Banditron

1 **Define**: $\Omega = \{W \in \mathbb{R}^{Kd} : \|\mathbf{e}_i^\top W\|_2 \le D\}$.
2 For a set S of W's, $S(i|x)$ is the subset of S that outputs class i given feature vector x, i.e., $S(i|x) = \{W \in S : (Wx)_i \ge (Wx)_j \ \forall j\}$
3 $|S|$ denotes the volumn of S.
4 **parameter**: $\alpha \in (0, 1)$
5 $\Omega_1 = \Omega$.
6 **for** $t = 1, \dots, T$ **do**
7 **if** $\operatorname{argmax}_i \frac{|\Omega_t(i|x_t)|}{|\Omega_t|} \ge 1 - \alpha$ **then**
8 Let $\tilde{y}_t = i$.
9 **if** $\tilde{y}_t \ne y_t$ **then**
10 $\Omega_{t+1} = \Omega_t \backslash \Omega_t(\tilde{y}_t|x_t)$.
11 **else**
12 Let $\tilde{y}_t \sim \text{unif}([K])$.
13 **if** $\tilde{y}_t = y_t$ **then**
14 $\Omega_{t+1} = \Omega_t(\tilde{y}_t|x_t)$.

Conjecture(should be true): If the volume of Ω_t becomes smaller than $\frac{|\Omega|}{N}$, then the algorithm won't make any error anymore. N should be in the order of $\Theta(\frac{1}{\gamma^{Kd}})$.

Rough analysis:
Each time the algorithm makes an error in Line 7, the volume becomes α times the original volume. So the algorithm will not make more than $\frac{\ln N}{\ln \frac{1}{\alpha}}$ mistakes in this case.

In the case of Line 10, $K \ln \frac{1}{\delta}$ errors will accompany with a $(1-\alpha)$-factor shrinkage in the volume. Therefore, the number of errors occurred in this case is upper bounded by $\frac{K \ln \frac{1}{\delta} \ln N}{\ln \frac{1}{1-\alpha}} \le \frac{K \ln \frac{1}{\delta} \ln N}{\alpha}$.

Now we discuss about the complexity. The main issue is how to maintain Ω_t. Each time the algorithm enters Line 10, Ω_t becomes more and more fragmented. But if Ω_t can be maintained with M convex cones, then Ω_{t+1} can be maintained with $(K-1)M \le KM$ convex cones. And we assume each cone's volume can be computed in poly(T) time. Each time the algorithm enters Line 14, the number of convex cones does not increase.

By the above discussion, there will be no more than $K^{\frac{\ln N}{\ln \frac{1}{\alpha}}}$ convex cones to maintain. And the error bound is in the order of $\frac{K \ln \frac{1}{\delta} \ln N}{\alpha}$ for some $\alpha < \frac{1}{2}$. Let's try to balance the number of errors and computational complexity. Let

$$
\begin{aligned}
& K^{\frac{\ln N}{\ln \frac{1}{\alpha}}} \approx \frac{K \ln \frac{1}{\delta} \ln N}{\alpha} \\
\Rightarrow & \frac{\ln N}{\ln \frac{1}{\alpha}} \ln K \approx \ln \left(K \ln \frac{1}{\delta} \ln N \right) + \ln \frac{1}{\alpha} \\
\Rightarrow & \text{pick } \ln \frac{1}{\alpha} = \sqrt{\ln N}.
\end{aligned}
$$

Thus the computational complexity is in the order of $K^{\sqrt{\ln N}} \times \text{poly}(T) = K^{\sqrt{Kd \ln \frac{1}{\delta}}}$. The error bound is $\mathcal{O}\left(e^{\sqrt{Kd \ln \frac{1}{\delta}}} K^2 d \ln \frac{1}{\delta} \ln \frac{1}{\gamma} \right)$.

Another viewpoint: let $\frac{1}{\alpha} = K^\beta$, then the complexity is $\left(\frac{1}{\gamma}\right)^{\frac{Kd}{\beta}} \times \text{poly}(T)$ and the error bound is $K^{\beta+1} \ln \frac{1}{\delta} \ln N$.

10 Gradient Descent [TODO]

Assumption 3. $\|x_t\|_2^2 \le 1$. *There is a* $W^* \in \mathcal{W}$ *such that* $\ell_t(W^*) \le 0$ *for all* t (ℓ_t *and* $\mathcal{W}$ *are defined below).*

Algorithm 6: Banditron

Input: $D \ge 2$, ϵ (picked in a later lemma).
Definition:

$$\begin{aligned}\ell_t(W) &\triangleq [1 - (Wx_t)_{y_t} + \max_{r \neq y_t}(Wx_t)_r]_+^2 \quad \text{(squared hinge loss)} \\ &= \Phi_t(Wx_t),\end{aligned}$$

where $\Phi_t(z) \triangleq [1 - \mathbf{e}_{y_t}^\top z + \max_{r \neq y_t} \mathbf{e}_r^\top z]_+^2$.
Also, define $\mathcal{W} = \{W \in \mathbb{R}^{K \times d} : \|\mathbf{e}_i^\top W\|_2 \le D \text{ for all } i \in [K]\}$.
Initialization: $W_1 = 0, M_1 = I$.
for $t = 1, \dots, T$ **do**
  Observe x_t.
  if $\|x_t\|_{M_t^{-1}} \ge \epsilon$ *and* $\|W_t - W^*\|_F \ge 1$ **then**
    Draw $\tilde{y}_t \sim \text{unif}([K])$.
  else
    Draw $\tilde{y}_t = \hat{y}_t \triangleq \text{argmax}_{r \in [K]}(W_t x_t)_r$.
  if $\tilde{y}_t = y_t$ **then**
    $Z_t \leftarrow 1$,
    $M_{t+1} \leftarrow M_t + Z_t \ell_t(W_t) x_t x_t^\top$,
    $W_{t+1} \leftarrow \Pi_{\mathcal{W}}(W_t - \eta_{t+1} \nabla \ell_t(W_t))$, where $\eta_{t+1} = \frac{1}{8}$.
    ($\Pi_{\mathcal{W}}$ is the projection operator onto $\mathcal{W}$ w.r.t. Frobenius norm)
  else
    $Z_t \leftarrow 0$,
    $M_{t+1} \leftarrow M_t$,
    $W_{t+1} \leftarrow W_t$.

Lemma 8. $\|\nabla \ell_t(W)\|_F^2 \le 8\ell_t(W)$.

Proof. $\|\nabla \ell_t(W)\|_F^2 = \|\nabla \Phi_t(Wx_t) x_t^\top\|_F^2 \le \left(2\sqrt{\Phi_t(Wx_t)}\right)^2 \times 2\|x_t\|_2^2 \le 8\ell_t(W)$. □

Lemma 9. *Let* $L_{t+1} \triangleq \sum_{s=1}^t Z_s \ell_s(W_s)$. *Then* $\|W_{t+1} - W^*\|_F^2 \le \exp\left(-\frac{L_{t+1}}{32KD^2}\right)$.

Proof. Let $Z_t = 1$.

$$\begin{aligned}\|W_{t+1} - W^*\|_F^2 &\le \|W_t - \eta_{t+1}\nabla \ell_t(W_t) - W^*\|_F^2 \\ &= \|W_t - W^*\|_F^2 - 2\eta_{t+1}\langle \nabla \ell_t(W_t), W_t - W^*\rangle_F + \eta_{t+1}^2 \|\nabla \ell_t(W_t)\|_F^2.\end{aligned}$$

By the separable assumption we have $\ell_t(W^*) \le 0$. Since ℓ_t is convex, $\langle \nabla \ell_t(W_t), W_t - W^*\rangle \ge \ell_t(W_t) - \ell_t(W^*) \ge \ell_t(W_t)$. Continuing the above calculation and using Lemma 8, we get

$$\begin{aligned}\|W_{t+1} - W^*\|_F^2 &\le \|W_t - W^*\|_2^2 - 2\eta_{t+1}\ell_t(W_t) + 8\eta_{t+1}^2 \ell_t(W_t) \\ &\le \|W_t - W^*\|_F^2 - \frac{1}{8}\ell_t(W_t) \\ &\le \|W_t - W^*\|_F^2 \left(1 - \frac{\ell_t(W_t)}{32KD^2}\right) \quad \text{because } \|W_t - W^*\|_F^2 \le 4KD^2 \\ &\le \|W_t - W^*\|_F^2 \exp\left(-\frac{\ell_t(W_t)}{32KD^2}\right)\end{aligned}$$

By induction, we can get

$$\|W_{t+1} - W^*\|_F^2 \le KD^2 \exp\left(-\frac{L_{t+1}}{32KD^2}\right)$$

□

Definition 10. $\|W\|_M^2 \triangleq \sum_{i=1}^K \left\|\mathbf{e}_i^\top W\right\|_M^2$.

With this definition we have $\|Wx_t\|_2^2 = \sum_{i=1}^K (\mathbf{e}_i^\top W x_t)^2 \le \sum_{i=1}^K \|\mathbf{e}_i^\top W\|_M^2 \|x_t\|_{M^{-1}}^2 \le \|W\|_M^2 \|x_t\|_{M^{-1}}^2$

Lemma 11.

$$\|W_t - W^*\|_{M_t}^2 \le (1+L_t)K^2D^2 \exp\left(-\frac{L_t}{32KD^2}\right) \le 32K^3D^4.$$

Proof. Because we assume $\|x_t\|_2^2 \le 1$, it holds that $M_t \preceq (1+L_t)I$. Therefore $\|W_t - W^*\|_{M_t}^2 \le (1+L_t)\|W_t - W^*\|_I^2 = (1+L_t)\sum_{i=1}^K \|\mathbf{e}_i^\top (W_t - W^*)\|_2^2 \le (1+L_t)K\|W_t - W^*\|_F^2$. By Lemma 9 this is bounded by $(1+L_t)K^2D^2 \exp\left(-\frac{L_t}{32KD^2}\right)$, which can further be bounded by a constant related to K and D. For example, using the property $\exp(-x) \le \frac{1}{(1+x)^2}$ for all $x > 0$, it can be upper bounded by $(1+L_t)K^2D^2 \times \frac{(32KD^2)^2}{(L_t+32KD^2)^2} \le \frac{32^2K^4D^6}{32KD^2+L_t} \le 32K^3D^4$. □

Lemma 12. *If* $\|x_t\|_{M_t^{-1}} \le \epsilon = \frac{1}{4D\sqrt{32K^3D^4}}$, *then* $\hat{y}_t = y_t$.

Proof. By the convexity of ℓ_t,

$$\begin{aligned}
\ell_t(W_t) \le \ell_t(W_t) - \ell_t(W^*) &\le \langle \nabla \ell_t(W_t), W_t - W^* \rangle \\
&= \left\langle \nabla \Phi_t(W_t x_t) x_t^\top, W_t - W^* \right\rangle \\
&= \langle \nabla \Phi_t(W_t x_t), W_t x_t - W^* x_t \rangle \\
&\le 4D \left\| W_t x_t - W^* x_t \right\|_2 \\
&\le 4D \left\| W_t - W^* \right\|_{M_t} \left\| x_t \right\|_{M_t^{-1}} \le 1.
\end{aligned}$$

This implies $\hat{y}_t = y_t$. □

Therefore, when we do not explore, we know W_t will predict correctly! Thus we only need to bound the number of errors occurred in exploration rounds, which is calculated by the following lemma.

Lemma 13. $\sum_{t=1}^T \mathbf{1}[\tilde{y}_t \ne y_t] \le$??? *with probability at least* $1-\delta$.

Proof. By the above discussion, $\sum_{t=1}^T \mathbf{1}[\tilde{y}_t \ne y_t] \le N \triangleq \sum_{t=1}^T Z_t$, the number of exploration rounds.

$$\begin{aligned}
N &= \sum_{t=1}^T \mathbf{1}\left[\|x_t\|_{M_t^{-1}} > \epsilon\right] \\
&\le \left(K \ln \frac{1}{\delta}\right) \sum_{t=1}^T \mathbf{1}\left[\|x_t\|_{M_t^{-1}} > \epsilon\right] Z_t \qquad \text{(when } \|x_t\|_{M_t^{-1}} \ge \epsilon, \tilde{y}_t = y_t \text{ with probability } \tfrac{1}{K}\text{)} \\
&\le \frac{K \ln \frac{1}{\delta}}{\epsilon^2} \sum_{t=1}^T \|x_t\|_{M_t^{-1}}^2 Z_t \le \max_{t \in [T]} \left(\frac{1}{\ell_t(W_t)}\right) \times \frac{K \ln T \ln \frac{1}{\delta}}{\epsilon^2}.
\end{aligned}$$

□

Discussion. In the calculation of Lemma 12, we can actually get $\ell_t(W_t)^2 \le \|\nabla \Phi_t(W_t x_t)\|_2^2 \|W_t - W^*\|_{M_t}^2 \|x_t\|_{M_t^{-1}}^2$. Similar to the calculation in Lemma 9, $\|\nabla \Phi_t(W_t x_t)\|_2^2$ is bounded by constant times $\ell_t(W_t)$. So the exploration criterion could potentially become $\ell_t(W_t)\|x_t\|_{M_t^{-1}}^2 \ge \frac{1}{\epsilon^2}$, which makes Lemma 13 go through. The problem is just we do not know $\ell_t(W_t)$ in general.

11 Continuous EXP4 with Uniform Exploration

Algorithm 7: Banditron

Parameters: feasible set $\Omega \subset \mathbb{R}^{K\times d}$

Definitions: $\ell_t(W) \triangleq [1-(Wx_t)_{y_t}+\max_{r\in[K]}(Wx_t)_r]_+$ (hinge loss)

for $t=1,\dots,T$ **do**

Receive $x_t \in \mathbb{R}^d$.

Define

$$q_t(W) = \frac{\exp(-\alpha\sum_{s=1}^{t-1}\hat{\ell}_s(W))}{\int_{U\in\Omega}\exp(-\alpha\sum_{s=1}^{t-1}\hat{\ell}_s(U))dU}, \quad \forall W\in\Omega,$$

where $\hat{\ell}_s(W) = \mathbf{1}[\tilde{y}_s = y_s]\left(\frac{\mathbf{1}[\hat{y}_s=y_s]\ell_s(W)}{1-\gamma+\frac{\gamma}{K}} + \frac{\mathbf{1}[\hat{y}_s\neq y_s]\ell_s(W)}{\frac{\gamma}{K}}\right)$.

Sample $W_t \sim q_t$, and let $\hat{y}_t = \operatorname{argmax}_{r\in[K]}(W_t x_t)_r$.

Let $\tilde{y}_t = \hat{y}_t$ with probability $1-\gamma$, and $\tilde{y}_t \sim \text{unif}([K])$ with probability γ.

Lemma 14. $\mathbb{E}_{\tilde{y}_t}[\hat{\ell}_t(W)] = \ell_t(W)$ *for all* W.

Proof.

$$\begin{aligned}\mathbb{E}_{\tilde{y}_t}[\hat{\ell}_t(W)] &= \mathbb{E}_{\tilde{y}_t}\left[\mathbf{1}[\hat{y}_s=y_s]\frac{\mathbf{1}[\tilde{y}_s=y_s]\ell_t(W)}{1-\gamma+\frac{\gamma}{K}} + \mathbf{1}[\hat{y}_s\neq y_s]\frac{\mathbf{1}[\tilde{y}_s=y_s]\ell_t(W)}{\frac{\gamma}{K}}\right]\\ &= \mathbf{1}[\hat{y}_s=y_s]\ell_t(W) + \mathbf{1}[\hat{y}_s\neq y_s]\ell_t(W) = \ell_t(W).\end{aligned}$$

□

Plugging these lemmas in the previous hedge bound, we can get

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^T \ell_t(W_t)\right]\\ &= \mathbb{E}\left[\sum_{t=1}^T \int_{W\in\Omega} q_t(W)\ell_t(W)dW\right]\\ &= \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_{\tilde{y}_t}\left[\int_{W\in\Omega} q_t(W)\hat{\ell}_t(W)dW\right]\right]\\ &\le \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_{\tilde{y}_t}\left[\hat{\ell}_t(W^*)\right] + \frac{\mathbf{Ent}(q_1||\delta(W^*))}{\alpha} + \alpha\sum_{t=1}^T \mathbb{E}_{\tilde{y}_t}\left[\int_{W\in\Omega} q_t(W)\hat{\ell}_t(W)^2dW\right]\right]\\ &\le \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_{\tilde{y}_t}\left[\hat{\ell}_t(W^*)\right] + \frac{\mathbf{Ent}(q_1||\delta(W^*))}{\alpha} + \frac{2K\alpha}{\gamma}\sum_{t=1}^T \mathbb{E}_{\tilde{y}_t}\left[\int_{W\in\Omega} q_t(W)\hat{\ell}_t(W)dW\right]\right]\end{aligned}$$

... to bound the regret, it would be something like bounding $\frac{1}{1-\frac{K\alpha}{\gamma}}\left(\frac{1}{\alpha}+\frac{K\alpha}{\gamma}L^*+\gamma T\right)$, which gives $(L^*T)^{1/3}+\sqrt{T}$ regret bound.

Discussion. We can change $\ell_t(\cdot)$ to any reasonable convex loss (e.g., logsitic loss or second-order loss).

References

[1] Alina Beygelzimer, Francesco Orabona, and Chicheng Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In *International Conference on Machine Learning*, 2017.

[2] Dylan J Foster, Satyen Kale, Haipeng Luo, Mehryar Mohri, and Karthik Sridharan. Logistic regression: The importance of being improper. *Proceedings of Machine Learning Research*, 75:1–42, 2018.

[3] Elad Hazan and Satyen Kale. Newtron: an efficient bandit algorithm for online multiclass prediction. In *Advances in neural information processing systems*, pages 891–899, 2011.

[4] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[5] Adam R Klivans and Rocco A Servedio. Learning intersections of halfspaces with a margin. In *International Conference on Computational Learning Theory*, 2004.