# Homework 3

CSE 541: Interactive Learning
Instructor: Kevin Jamieson
Due 11:59 PM on June 8, 2021 (late homework not accepted)

**Martingale analysis**

1. Let $f : \mathcal{K} \to \mathbb{R}$ be a convex function that is $G$-Lipschitz over a bounded, closed, convex set $\mathcal{K} \subset \mathbb{R}^d$ and assume $\nabla f(x)$ exists for all $x \in \mathcal{K}$. Assume that $\mathcal{K}$ has diameter at most $R$, i.e., $\|x - y\| \leq R$ for all $x, y \in \mathcal{K}$. You are given access to a stochastic first-order oracle, which at each point $x_t \in \mathcal{K}$ returns a stochastic gradient $\tilde{g}_t$ satisfying:

$$\mathbb{E}[\tilde{g}_t \mid x_t] = \nabla f(x_t), \quad \|\tilde{g}_t\| \leq G \text{ almost surely.}$$

Consider the projected stochastic gradient descent (SGD) algorithm:

- Initialize $x_1 \in \mathcal{K}$
- For $t = 1, \ldots, T$: update
$$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta \tilde{g}_t)$$
where $\Pi_{\mathcal{K}}$ denotes Euclidean projection onto $\mathcal{K}$ and $\eta > 0$ is a fixed step size.

Let $\bar{x}_T := \frac{1}{T}\sum_{t=1}^T x_t$ be the average iterate. In this problem, you will derive a high-probability bound on the suboptimality gap $f(\bar{x}_T) - f(x^*)$, where $x^* \in \arg\min_{x \in \mathcal{K}} f(x)$. In class we showed that

$$\sum_{t=1}^T \langle \tilde{g}_t, x_t - x^* \rangle \leq \frac{R^2}{2\eta} + \frac{\eta G^2 T}{2}.$$

*(a) Suboptimality Decomposition*

Let $g_t := \mathbb{E}[\tilde{g}_t \mid x_t] = \nabla f(x_t)$, and use the decomposition $\tilde{g}_t = g_t + (\tilde{g}_t - g_t)$ to write:

$$\sum_{t=1}^T \langle g_t, x_t - x^* \rangle \leq \frac{R^2}{2\eta} + \frac{\eta G^2 T}{2} - \sum_{t=1}^T \langle \tilde{g}_t - g_t, x_t - x^* \rangle.$$

Argue that by convexity of $f$,

$$\sum_{t=1}^T f(x_t) - f(x^*) \leq \frac{R^2}{2\eta} + \frac{\eta G^2 T}{2} + \sum_{t=1}^T Z_t,$$

where $Z_t := -\langle \tilde{g}_t - g_t, x_t - x^* \rangle$.

*(b) Martingale Concentration*

Show that $(Z_t)$ is a martingale with respect to the filtration $\mathcal{F}_t = \sigma(x_1, \ldots, x_t, \tilde{g}_1, \ldots, \tilde{g}_{t-1})$, and that $|Z_t| \leq 2GR$. Using Hoeffding's lemma, show that $M_t(\lambda) = \exp\left(\lambda S_t - t\lambda^2/(4GR)\right)$ is a supermartingale, where $S_t = \sum_{s=1}^t Z_t$. Then show that with probability at least $1 - \delta$:

$$\sum_{t=1}^T Z_t = S_t \leq \sqrt{8G^2 R^2 T \log(1/\delta)}.$$

*(c) Conclude the Bound*

Using Jensen's inequality, combine the above to show that with probability at least $1 - \delta$:

$$f(\bar{x}_T) - f(x^*) \leq \frac{1}{T}\left(\frac{R^2}{2\eta} + \frac{\eta G^2 T}{2} + \sqrt{8G^2 R^2 T \log(1/\delta)}\right).$$

Choosing the optimal fixed step size $\eta = \frac{D}{G\sqrt{T}}$, we conclude that:

$$f(\bar{x}_T) - f(x^*) \leq \frac{GD}{\sqrt{T}} \left(1 + 2\sqrt{2\log(1/\delta)}\right)$$

with probability at least $1 - \delta$.

**Non-stochastic Bandits**
Using problem 5 from homework 1, repeat those experiments but add EXP3 and instead of using a Gaussian distribution with mean $\mu_i$, use the distribution .75 with probability $(1+\mu)/2$ and $-.75$ with probability $(1-\mu)/2$. Try different values of $\gamma$ and $\eta$ for EXP3. No need to change the Thompson sampling algorithm (i.e., use the Gaussian update even though we're using Bernoulli's now).
**Contextual Bandits**
3. Problem 18.8 of [SzepesvariLattimore].

4. In this exercise we will implement several contextual bandit algorithms. We will "fake" a contextual bandit problem with multi-class classification dataset where each example is context, and the learner chooses an "action" among the available class labels, and receives a reward of 1 if the guess was correct, and 0 otherwise. However, keeping with bandit feedback, we assume the learner only knows the reward of the action played, not all actions.

We will use the MNIST dataset[1]. The MNIST dataset contains 28x28 images of handwritten digits from 0-9. Download this dataset and use the python-mnist library[2] to load it into Python. Rather than using the full images, you may run PCA on the data to come up with a lower dimensional representation of each image. You will have to experiment with what dimension, $d$, to use. Scale all images so that they are norm 1.

Let the $d$ dimensional representation of the $t$th image in the dataset, $c_t$, be our "context." Our action set $\mathcal{A} = \{0, 1, \ldots, 9\}$ has 10 actions associated with each label. For each $i \in \mathcal{A} = \{0, 1, \ldots, 9\}$ define the feature map $\phi(c, i) = \text{vec}(c\mathbf{e}_i^\top) \in \mathbb{R}^{10d}$. If $v(c, a)$ is the expected reward of playing action $a \in \mathcal{A}$ in response to context $c$, then let us "model the world" with the simple linear model so that $v(c, a) \approx \langle \theta_*, \phi(c, a) \rangle$ for some unknown $\theta_* \in \mathbb{R}^{10d}$. Of course, when actually playing the game we will observe image features $c_t$ as the context, choose an "action" $a_t \in \{0, \ldots, 9\}$, and receive reward $r_t = \mathbf{1}\{a_t = y_t\}$ where $y_t$ is the true label of the image $c_t$ and $a_t$ is the action played.

Implement the Explore-Then-Commit algorithms, Follow-The-Leader, LinUCB, and Thompson Sampling algorithms for this problem. You can use just the training set of $T = 50000$ examples. The training set is class balanced meaning that there are 5000 examples of each digit. Important: randomly shuffle the dataset so the probability of any particular class showing up at any given time is 1/10. The algorithms work as follows:

- **Explore-Then-Commit** ("Model the world"): Fix $\tau \in [T]$. For the first $\tau$ steps, select each action $a \in \mathcal{A}$ uniformly at random. Compute $\widehat{\theta} = \arg\min_\theta \sum_{t=1}^{\tau} (r_t - \langle \phi(c_t, a_t), \theta \rangle)^2$. For $t > \tau$ play $a_t = \arg\max_{a \in \mathcal{A}} \langle \phi(c_t, a), \widehat{\theta} \rangle$. Choose a value of $\tau$ and justify it.

- **Explore-Then-Commit** ("Model the bias"): Fix $\tau \in [T]$. For the first $\tau$ steps, select each action $a \in \mathcal{A}$ uniformly at random. Our goal is to identify a policy $\widehat{\pi} : \mathcal{C} \to \mathcal{A}$ using the dataset $\{(c_t, a_t, p_t, r_t)\}_{t \leq \tau}$ such that

$$\begin{aligned} \widehat{\pi} &= \arg\max_{\pi \in \Pi} \sum_{t=1}^{\tau} \frac{r_t \mathbf{1}\{\pi(c_t) = a_t\}}{p_t} \\ &= \arg\min_{\pi \in \Pi} \sum_{t=1}^{\tau} \frac{r_t \mathbf{1}\{\pi(c_t) \neq a_t\}}{p_t} \\ &= \arg\min_{\pi \in \Pi} \sum_{t \in [\tau] : r_t = 1} \mathbf{1}\{\pi(c_t) \neq a_t\} \end{aligned}$$

[1] http://yann.lecun.com/exdb/mnist/
[2] https://pypi.org/project/python-mnist/

where the last line uses the fact that $p_t = 1/10$ due to uniform exploration and the definition of $r_t$. Note that this is just a multi-class classification problem on dataset $\{(c_t, a_t)\}_{t\in[\tau]:r_t=1}$ where one is trying to identify a classifier $\widehat{\pi} : \mathcal{C} \to \mathcal{A}$ that predicts label $a_t$ from features $c_t$. Train a 10-class linear logistic classifier[3] $\widehat{\pi}$ on the data up to time $[\tau]$ and then for $t > \tau$ play $a_t = \arg\max_{a\in\{0,\ldots,9\}} \widehat{\pi}(c_t)$. Choose the same value of $\tau$ as "Model the world".

- **Follow-The-Leader**: Fix $\tau \in [T]$. For the first $\tau$ steps, select each action $a \in \mathcal{A}$ uniformly at random. For $t > \tau$ play $a_t = \arg\max_{a\in\mathcal{A}} \langle \phi(c_t, a), \widehat{\theta}_{t-1} \rangle$ where $\widehat{\theta}_t = \arg\min_\theta \sum_{s=1}^t (r_s - \langle \phi(c_s, a_s), \theta \rangle)^2$. Choose a value of $\tau$ and justify it.

- **LinUCB** Using Ridge regression with an appropriate $\gamma > 0$ ($\gamma = 1$ may be okay) construct the confidence set $\mathcal{C}_t$ derived in class (and in the book). At each time $t \in [T]$ play $a_t = \arg\max_{a\in\mathcal{A}} \max_{\theta\in\mathcal{C}_t} \langle \theta, \phi(c_t, a) \rangle$.

- **Thompson Sampling** Fix $\gamma > 0$ ($\gamma = 1$ may be okay). At time $t \in [T]$ draw $\widetilde{\theta}_t \sim \mathcal{N}(\widehat{\theta}_{t-1}, V_{t-1}^{-1})$ and play $a_t = \arg\max_{a\in\mathcal{A}} \langle \widetilde{\theta}_t, \phi(c_t, a) \rangle$ where $\widehat{\theta}_t = \arg\min_\theta \sum_{s=1}^t (r_s - \langle \theta, \phi(c_s, a_s) \rangle)^2$ and $V_t = \gamma I + \sum_{s=1}^t \phi(c_s, a_s)\phi(c_s, a_s)^\top$.

Implement each of these algorithms and show a plot of the regret (all algorithms on one plot) when run on MNIST for good choices of $\tau, \gamma$. Hint, for computing $V_t^{-1}$ efficiently see `https://en.wikipedia.org/wiki/Sherman%E2%80%93Morrison_formula`.

[3] Please feel free to use an off-the-shelf method to train logistic regression such as `https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html#sphx-glr-auto-examples-linear-model-plot-iris-logistic-py`