Advanced Topics in Machine Learning 2015-2016

Yevgeny Seldin Christian Igel Brian Brost

Home Assignment 4

Deadline: Sunday, 4 October, 2015, 23:59

The assignments must be answered individually - each student must write and submit his/her own solution. We encourage you to work on the assignments on your own, but we do not prevent you from discussing the questions in small groups. If you do so, you are requested to list your group partners in your individual submission.

Submission format: *Please, upload your answers in a single* `.pdf` *file and additional* `.zip` *file with all the code that you used to solve the assignment. (The* `.pdf` *should* ***not*** *be part of the* `.zip` *file.)*

IMPORTANT: *We are interested in how you solve the problems, not in the final answers. Please, write down all your calculations.*

Question 1 (Tighter analysis of the Hedge algorithm - 25 points)**.** Apply Hoeffding's lemma (Lemma 5 in "Machine Learning" handouts) in order to derive a better parametrization and a tighter bound on the expected regret of the Hedge algorithm. Guidance:

1. Traverse the analysis of Hedge algorithm that we did in class. There will be a place where you will have to bound expectation of an exponent of a function. Instead of going the way we did, apply Hoeffding's inequality.

2. Find the value of η that minimizes the new bound. (You should get $\eta = \sqrt{\frac{8 \ln N}{T}}$ - please, prove this formally.)

3. At the end you should obtain $R_T \leq \sqrt{\frac{1}{2} T \ln N}$. (I.e., you will get an improvement by a factor of 2 compared to what we did in class.)

Remark: Note that the regret upper bound matches the lower bound up to a constant. This is an extremely rare case.

Question 2 (Prediction of i.i.d. sequences - 25 points)**.** Assume that you have to predict a binary sequence $X_1, X_2, \ldots$ and that you know that X_i-s are i.i.d. Bernoulli variables with unknown bias μ. (You know that X_i-s are i.i.d., but you do not know the value of μ.) On every round you can predict "0" or "1" and your loss is the zero-one loss depending on whether your prediction matches the outcome. The regret is computed with respect to the performance of two experts - one that always predicts "0" and one that always predicts "1".

1. Suggest an algorithm for predicting the sequence that exploits the knowledge that the sequence is i.i.d. Bernoulli.

2. Write a simulation that compares numerically the performance of your algorithm with the performance of Hedge algorithm that we presented in class (with $\eta = \sqrt{\frac{2 \ln N}{T}}$) and the performance of reparametrized Hedge algorithm from Question 1 (with $\eta = \sqrt{\frac{8 \ln N}{T}}$). The Hedge algorithm should operate with the aforementioned two experts. To make things more interesting we will add "anytime" version of Hedge to the comparison. "Anytime" algorithm is an algorithm that does not depend on the time horizon. Let t be a running time index ($t = 1, \ldots, T$). Anytime Hedge corresponding to the simple analysis uses $\eta_t = \sqrt{\frac{\ln N}{t}}$ and anytime Hedge corresponding to the

tighter analysis in Question 1 uses $\eta_t = 2\sqrt{\frac{\ln N}{t}}$ (the learning rate η_t of anytime Hedge changes with time and does not depend on the time horizon). Some instructions for the simulation:

- Take time horizon $T = 1000$. (In general, time horizon should be large in comparison to $\frac{1}{(\mu-\frac{1}{2})^2}$.)
- Test several values of μ. We suggest $\mu = \frac{1}{2} - \frac{1}{4}$, $\mu = \frac{1}{2} - \frac{1}{8}$, $\mu = \frac{1}{2} - \frac{1}{16}$.
- Plot regret of the five algorithms with respect to the best out of "0" and "1" experts as a function of t for the different values of μ (make a separate plot for each μ). Make 10 runs of each algorithm and report the average regret over the 10 runs and the average regret + one standard deviation over the 10 runs. Do not forget to add a legend to your plot.

3. Which values of μ are more difficult for your algorithm?

4. Design an adversarial (non-i.i.d.) sequence on which you expect your algorithm to perform poorly. Explain the design of your adversarial sequence and report a plot with a simulation, where you compare the performance of your algorithm with the different versions of Hedge. As before, make 10 repetitions of the experiment and report the average and the average + one standard deviation. Comment on your observations.

Question 3 (The doubling trick - 25 points)**.** Consider the following forecasting strategy ("doubling trick"): time is divided into periods $(2^m, \ldots, 2^{m+1} - 1)$, where $m = 0, 1, 2, \ldots$. (In other words, the periods are $(1), (2, 3), (4, \ldots, 7), (8, \ldots, 15), \ldots$.) In period $(2^m, \ldots, 2^{m+1} - 1)$ the strategy uses the optimized Hedge forecaster from Question 1 initialized at time 2^m with parameter $\eta_m = \sqrt{\frac{8 \ln N}{2^m}}$. Thus, the Hedge forecaster is reset at each time instance that is an integer power of 2 and is restarted with a new value of η. By Question 1 we know that the regret of Hedge with $\eta_m = \sqrt{\frac{8 \ln N}{2^m}}$ within the period $(2^m, \ldots, 2^{m+1} - 1)$ is bounded by $\sqrt{\frac{1}{2} 2^m \ln N}$.

1. Prove that for any $T = 2^m - 1$ the overall regret (considering time period $(1, \ldots, T)$) of this forecasting strategy satisfies
$$R_T \leq \frac{1}{\sqrt{2} - 1} \sqrt{\frac{1}{2} T \ln N}.$$
(Hint: at some point in the proof you will have to sum an infinite geometric series. If you have never done it before - check on wikipedia or ask Brian how to do that.)

2. Prove that for any arbitrary time T the regret of this forecasting strategy satisfies
$$R_T \leq \frac{\sqrt{2}}{\sqrt{2} - 1} \sqrt{\frac{1}{2} T \ln N}.$$

Remark: The regret of "anytime" Hedge with $\eta_t = 2\sqrt{\frac{\ln N}{t}}$ satisfies $R_T \leq \sqrt{T \ln N}$ for any T. For comparison, $\frac{1}{\sqrt{2}(\sqrt{2}-1)} \approx 1.7$ and $\frac{1}{\sqrt{2}-1} \approx 2.4$. Thus, anytime Hedge is both more elegant and more efficient than Hedge with the doubling trick.

Question 4 (Value function - 25 points)**.** The floor plan in Figure 1 represents a 3×4 grid-world. Each of the 12 rooms corresponds to one state. The agent can go from one room to another if the rooms are connected by a door. The actions are the movements $\{\text{up}, \text{down}, \text{right}, \text{left}\}$. Every movement (except in the terminal state) gives a negative reward of -1 (i.e., every movements costs 1). If you bump into a wall without a door, you stay in the same room, but the movement has still to be paid. There are three rooms that give you an *additional* negative reward of -5 and -10, respectively, *when you enter them*, see figure. The room in the bottom right is a terminal state. An episode ends when this room is reached, which can be modelled by every action in this state leaving the state unchanged and having no cost.

This exercise requires an implementation of the MDP. Note that all rewards and transitions are deterministic. They could be described by simple mappings $S \times A \to \mathbb{R}$ and $S \times A \to S$, respectively, which can be encoded by simple tables.

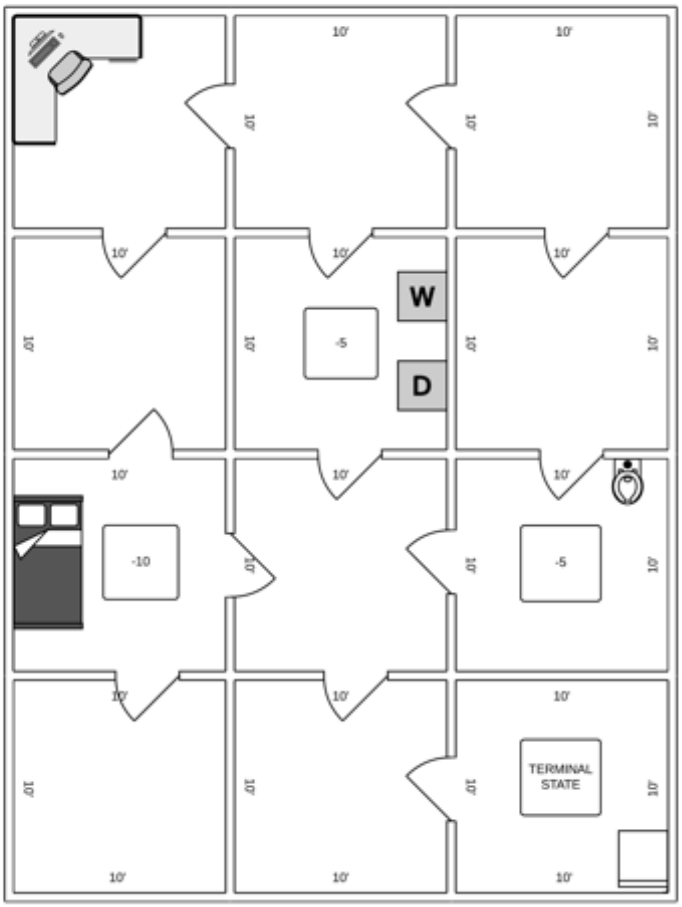

Figure 1: Illustration for Question 4.

Use an algorithm presented in the lecture to compute the value function V^{rand} of the random policy, that is, the policy that chooses an action uniformly at random in every state. Report the 12 V^{rand} values. Provide the implementation of the algorithm.

Use an algorithm presented in the lecture to compute the optimal value function V^*. Report the 12 V^* values. Provide the implementation of the algorithm.

Good luck!
Yevgeny, Christian, & Brian