MLRG Summer 2019

Online Learning and Applications

Yihan Zhou(Joey)

June 19th, 2019

The University of British Columbia

New Term, New Format

- Eight different topics in online learning.

New Term, New Format

- Eight different topics in online learning.

- Two parts—Theory+Application.

Topics

Week #	Topic	Paper 1	Application	Paper 2
1	Multiplicative Weight Update	"On-line algorithms in machine learning" by Blum "The Multiplicative Weights Update Method: a. Meta Algorithm and Applications" by Arora et al	Adaboost	"Lifelong Learning with Weighted Majority Voter" by Pentina and Urner
2	Follow the Leader	"Efficient algorithms for online decision problems" by Kalai and Vempala	Deep Learning	"Follow the moving leader in deep learning" by Zhang and Kwok
3	Intro to Bandits, UCB	"Regret analysis of stochastic and nonstochastic multi-armed bandit problems" (Chapter 1,2) by Bubeck et al	Ranking	"Learning Diverse Rankings with Multi-Armed Bandits" by Radlinski et al
4	Contextual Bandits	"Regret analysis of stochastic and nonstochastic multi-armed bandit problems" (Chapter 4) by Bubeck et al	News Recommendation	"A Contextual-Bandit Approach to Personalized News Article Recommendation" by Li et al
5	Thompson Sampling	"Linear Thompson Sampling Revisited" by Abeille and Lazaric	Contextual Bandits	"Thompson Sampling for Contextual Bandits with Linear Payoffs" by Agrawal and Goyal
6	Markovian Bandits	"Bandit Processes and Dynamic Allocation Indices" by Gittins; "Online Algorithms for the Multi-Armed Bandit Problem with Markovian Rewards" by Tekin and Liu	Stochastic Scheduling	"Stochastic Scheduling" by Nino-Mora
7	Dueling Bandits	"The K-armed dueling bandits problem" by Yue et al	Information Retrieval	"Interactively optimizing information retrieval systems as a dueling bandits problem" By Yue and Joachims
8	Linear Bandits - Online linear optimization	"Regret analysis of stochastic and nonstochastic multi-armed bandit problems" (Chapter 5) by Bubeck et al	Adaptive Routing	"Online linear optimization and adaptive routing" by Awerbuch and Kleinberg

Overview of Topics

Predicting from Expert Advice

- Input of the algorithm: Advices from n "experts".

Predicting from Expert Advice

- Input of the algorithm: Advices from n “experts“.
- After the prediction, the right answer is revealed.

Predicting from Expert Advice

- Input of the algorithm: Advices from n “experts“.
- After the prediction, the right answer is revealed.
- No assumptions of quality or independence of the experts.

Predicting from Expert Advice

- Input of the algorithm: Advices from n “experts“.
- After the prediction, the right answer is revealed.
- No assumptions of quality or independence of the experts.
- Goal: Perform nearly as well as the best expert so far.

Weighted Majority

The Weighted Majority Algorithm (simple version)

1. Initialize the weights $w_1, \cdots, w_n$ of all the experts to 1.
2. Given a set of predictions $\{x_1, \cdots, x_n\}$ by the experts, output the prediction with the highest total weight. That is, output 1 if

$$\sum_{i:x_i=1} x_i \geq \sum_{i:x_i=1} x_i$$

and output 0 otherwise.

3. When the correct answer l is received, penalize each mistaken expert by multiplying its weight by $1/2$. That is, if $x_i \neq l$, then $w_i \leftarrow w_i/2$; if $x_i = l$ then w_i is not modified. Goto 2.

Extension of Framework

- Decisions(predictions) may not be binary or even discrete.

Extension of Framework

- Decisions(predictions) may not be binary or even discrete.
- Define a cost function $f_t(w_t)$ for decision w_t made in each round t.

Extension of Framework

- Decisions(predictions) may not be binary or even discrete.
- Define a cost function $f_t(w_t)$ for decision w_t made in each round t.
- Our goal is to ensure that our total cost is not much larger than the minimum total cost of any expert, i.e.,

$$\min_{w_1,\cdots,w_T} \left(\sum_{t=1}^{T} f_t(w_t) - \min_{w} \sum_{t=1}^{T} f_t(w) \right).$$

  The objective is called regret.

Multiplicative Weights Update

Probabilistic Experts Algorithm (simple version)

1. Initialize $u_i = 1$, where u_i is the weight of the i-th expert.
2. Predict according to an expert chosen with probability proportional to u_i, the probability of choosing the i-th expert is u_i/U where U is the total weight.
3. Update weights by setting $u_i \leftarrow u_i(1-\epsilon)^{f_t(w_i)}$ for all experts.

Follow the Leader

A more intuitive algorithm is called follow the leader(FTL),

$$w_t = \operatorname*{argmin}_{w} \sum_{i=1}^{t} f_i(w)$$

There are some variants:

- Adding regularization, follow the regularized leader(FTRL)
- Adding perturbation, follow the perturbed leader(FTPL)

Bandits Setting

The bandits setting is different from the prediction from expert advice setting in:

- Instead of incurring costs, we get "rewards"! So the regret becomes

$$\max_{w} \sum_{t=1}^{T} f_t(w) - \sum_{t=1}^{T} f_t(w_t).$$

Bandits Setting

The bandits setting is different from the prediction from expert advice setting in:

- Instead of incurring costs, we get "rewards"! So the regret becomes

$$\max_{w} \sum_{t=1}^{T} f_t(w) - \sum_{t=1}^{T} f_t(w_t).$$

- Key difference: The algorithm observes only the reward for the selected action, and nothing else. This is called bandit feedback.

Stochastic Bandits

- The rewards of each action(decision) a is i.i.d. according to some distribution $\mathcal{D}_a$.
- The regret is

$$\max_w \mathbb{E}\left[f(w)\right] \cdot T - \sum_{t=1}^{T} \mathbb{E}\left[f(w_t)\right].$$

- Exploration exploitation trade-off.

Exploration Strategy

- Non-adaptive exploration: uniform exploration, ϵ-greedy

- Adaptive exploration: successive elimination, optimism under uncertainty.
  - Upper confidence bound(UCB):

$$\text{UCB}_t(a) = \bar{\mu}_t(a) + r_t(a)$$

    $\bar{\mu}_t(a)$ is the average reward of action a up to round t and $r_t(a) = \sqrt{\frac{2 \log t}{n_t(a)}}$, where $n_t(a)$ denotes the number of times arm a gets pulled till round t.
  - UCB stragety: Pull every arm once and pick actions according to UCB.

Thompson Sampling

- Draw an arm from the posterior distribution p_t of the best arm a^*,

$$p_t(a) = P[a = a^* | H_t] \text{ for each arm } a.$$

  where $H_t = \{(a_1, r_1), \cdots, (a_t, r_t)\}$.
- From another perspective, Thompson sampling sample reward function μ_t from the posterior distribution and choose the best arm according to μ_t.

Other Bandits

- Contextual bandits: Rewards in each round depend on a context, which is observed by the algorithm prior to making a decision.
- Markovian bandits: Rewards are Markovian. The ith arm is modeled as an irreducible Markov chain with finite state space S^i.
- Dueling bandits: Each iteration comprises of a noisy comparison (a duel) between two bandits (possibly the same bandit with itself).
- Linear bandits: The set of arms $\mathcal{K}$ is a compact set, not necessarily discrete. The reward function is assumed to be linear.

References

This presentation uses content from Blum's survey, Slivkins's new textbook of bandits, Bubeck's textbook of bandits and papers from our list.

Call for volunteers for presentation!

Thank you!