Lecture 28: Last Lecture

SUMMARY

1. Statistical learning (supervised):
   - Rates: Rademacher complexity, covering number, VC dimension
   - Algorithm : Empirical Risk Minimization (ERM)
2. Online learning (supervised):
   - Rates: sequential Rademacher complexity, Equiv. Seq. versions: seq. covering number, Littlestone dimension
   - Algorithm : relaxation based approach ($\mathbf{Rad}_n$ is universal)
   - Practical Algorithms: via relaxations, via online convex optimization, more . . . Burkholder method
3. $\approx$ Except for learning thresholds and halfspaces: online and statistical rates match
4. O2B: Use online learning algorithms for batch learning

SUMMARY : OF THINGS WE SKIPPED

1. There are deeper connections :
   - Between iid complexity tools and iid empirical process theory (uniform law of large numbers)
   - Between sequential complexity tools and uniform martingale law of large numbers
   - Applications: concentration results in Banach spaces and concentration of functions of RVs
2. Between these tools (for linear classes) and convex optimization and stochastic convex optimization

Whirlwind Tour: Itinerary

1. Other Partial Information Problems
2. Learning and game theory
3. Learning and optimization
4. Online control (full information)

Other partial information problems

1. Partial monitoring: Fixed loss matrix L and observation matrix H $(M \times N)$.
   - We pick action $i_t \in [M]$
   - Adversary picks action $j_t \in [N]$
   - We pay loss $L[i_t, j_t]$ but only observe $H[i_t, j_t]$
   - Rates either $n^{-1/2}$ or $n^{-1/3}$ or $O(1)$ based on conditions on H and L

2. Online zeroth order convex Lipchitz optimization:
   - Loss is convex and Lipschitz in out action (not just linear)
   - Key idea : Unbiased estimate of gradient

$$\nabla L_t(\hat{\mathbf{y}}_t) = \mathbb{E}\left[d \frac{\hat{\mathbf{y}}_t + \delta \mathbf{u}_t}{\delta} \mathbf{u}_t \right]$$

   - Open problem: $\Theta(n^{-1/2})$ proved 3 years or so ago.

Learning and Games: Zero-sum two player games

- Payoff matrix L of size $M \times N$
- Player I plays a regret minimizing strategy and player 2 just plays best-response, average pay off approximately as good as minimax value
- If both players play regret minimizing strategy, average moves converge to minimax equilibria

Learning and Games: Zero-sum two player games

$$\frac{1}{n}\sum_{t=1}^{n} a_t^\top L b_t \le \min_{a\in\Delta_M} \frac{1}{n}\sum_{t=1}^{n} a^\top L b_t + \sqrt{\frac{\log M}{n}}$$

$$\le \sup_{b\in\Delta_N} \min_{a\in\Delta_M} a^\top L b + \sqrt{\frac{\log M}{n}}$$

$$\le \text{minimax value} + \sqrt{\frac{\log M}{n}}$$

Learning and Games: Zero-sum two player games

- But faster rate of $O(\log M + \log N/n)$ possible if both player adapt to the other playing learning algorithm
- First cut idea: if other player is playing regret minimization, her/his next move is close to previous and vice versa: adapting to this we can speed up
- Second cut: an algorithm called optimistic mirror descent gets the right rate with a slightly more rigorous analysis
- Recent work shows that a whole lot of convex optimization algorithms can be derived by converting them into two player games and having simple algorithms play against each other

Learning and Games: Multi-player games

- Correlated equilibrium: Joint distribution q^* on moves of players is said to be a correlated equilibrium if for every player $i \in [K]$, and any $\pi_i : [N_i] \mapsto [N_i]$

$$\mathbb{E}_{(a_1,\ldots,a_K)\sim q^*}\left[L_i(a_i, a^{-i})\right] \leq \mathbb{E}_{(a_1,\ldots,a_K)\sim q^*}\left[L_i(\phi_i(a_i), a^{-i})\right]$$

- Minimize swap regret (mixed action) for each player:

$$\frac{1}{n}\sum_{t=1}^{n} L_i(a_t, a_t^{-i}) - \inf_{\phi} \frac{1}{n}\sum_{t=1}^{n} L_i(\phi(a_t), a_t^{-i}) \to 0$$

- The average moves of players converge to correlated equilibrium
- Generalize to any set of mappings Φ

Learning and Optimization: Convex optimization

- Regret minimization strategy applied to same function for the n steps provides an optimization algorithm
- Oracle efficiency
  1. Efficiency measured by counting number of local oracle calls to get desired sub-optimality level
  2. Eg. number of gradient calculations, function evaluations, higher derivatives etc.
- $\mathcal{F}$ is a d dimensional convex set: convex Lipchitz function
  - Efficiency bounded by $d \log(1/\epsilon)$
  - What about when d is large, say order $1/\epsilon^2$
  - MD and online methods optimal: proof through seq. Rademacher
- Statistical learning is stochastic convex optimization

Learning and Optimization: approximation algorithms

- Consider problem of form

$$\underset{f \in \mathcal{G}}{\text{argmax}} \quad c^\top f$$
$$\forall i \in [d] \;\; G_i(f) \leq 1$$

  G_i's are convex and easy to optimize over $\mathcal{G}$ (Eg. linear)
- Say F^* the value of the above (not known, do binary search)
- Say $\mathcal{G}$ is an easy polytope (Eg. linear equalities, flow polytope)
- Define $\mathcal{F} = \{f \in \mathcal{G} : c^\top f = F^*\}$
- Solution can be seen as:

$$f^* = \underset{f \in \mathcal{F}}{\text{argmin}} \max_{i \in [d]} G_i(f)$$

Learning and Optimization: approximation algorithms

- Run experts algorithm over d experts (for the d constraints) and returns $q_t \in \Delta_d$ a distribution over (negative) constraints
- On each round return $f_t = \mathsf{argmin}_{f \in \mathcal{F}} \mathbb{E}_{i_t \sim q_t} [G_{i_t}(f)]$ (assume this is easy to do)
- We claim $\bar{f}_n = \frac{1}{n} \sum_{t=1}^{n} f_t$ is a solution that only mildly violates constraints and obtains optimal value F^*.
- Now if we can find an $f_0 \in \mathcal{G}$ s.t. $c^\top > 0$ and $\forall i \in [d],\ G_i(f_0) \leq 1 - \gamma$
- Then for appropriate α, $\hat{f}_t = (1-\alpha)\bar{f}_n + \alpha f_0$ is a $(1-\epsilon)$ approximate solution and satisfies constraints

Learning and Optimization: approximation algorithms

Regret bound implies:

$$\begin{aligned}
\max_{i\in[d]} \frac{1}{n}\sum_{t=1}^{n} G_i(f_t) &\le \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}_{i_t\sim q_t}\left[G_{i_t}(f_t)\right] + \sqrt{\frac{\log d}{n}} \\
&= \frac{1}{n}\operatorname*{argmin}_{f_t\in\mathcal{F}} \sum_{t=1}^{n} \mathbb{E}_{i_t\sim q_t}\left[G_{i_t}(f_t)\right] + \sqrt{\frac{\log d}{n}} \\
&\le \frac{1}{n}\sum_{t=1}^{n} \mathbb{E}_{i_t\sim q_t}\left[G_{i_t}(f^*)\right] + \sqrt{\frac{\log d}{n}} \\
&\le 1 + \sqrt{\frac{\log d}{n}}
\end{aligned}$$

Learning and Optimization: approximation algorithms

Regret bound implies:

$$1 + \sqrt{\frac{\log d}{n}} \geq \max_{i \in [d]} \frac{1}{n} \sum_{t=1}^{n} G_i(f_t)$$
$$\geq \max_{i \in [d]} G_i \left(\frac{1}{n} \sum_{t=1}^{n} f_t \right)$$

Choosing n the number of iterations to run to appropriately with ϵ gives us a solution that ϵ-violates the constraints

Learning and Optimization: approximation algorithms

- This approach was key to getting $1 - \epsilon$ approximate max-flow solution in time

$$\tilde{O}\left(|E|\,|V|^{1/3}\epsilon^{-11/3}\right)$$

  which was subsequently improved to near linear time in $|E|$.
- One can also replace the optimization of f_t in each time step by a regret minimization algorithm

Online Control

- State variables in state space $\mathcal{S}$, loss depends on state.
- For $t = 1$ to n
  - Learner receives state information $s_t = D(s_{t-1}, \hat{y}_{t-1})$
  - Learner picks control $\hat{y}_t$
  - Adversary picks loss ℓ_t

Goal: Minimize policy regret

$$\sum_{t=1}^{n} \ell_t(s_t, \hat{y}_t) - \inf_{u \in \mathcal{U}} \sum_{t=1}^{n} \ell_t(s_t^u, u)$$

where $s_t^u = D(s_{t-1}, u)$ (states for comparator evolve according to comparators choice of action)

Online Control

- If dynamics is a contractive mapping then $s_t^u = D(s_{t-1}, u) \to s_*^u$.
- If algorithm choosing $\hat{y}_t$ is slow moving then states don't change drastically
- Hence we can think about the problem as online learning with losses fixed at loss at state where we play current action repeatedly for past few rounds.
- This gives us a reduction to online learning provided the algorithm we use is slow moving

Reinforcement Learning

A whole lot of exciting developments, check out

CS 6789: Foundations of Reinforcement Learning