Lecture 11

Huang Ziheng

November 30, 2017

1 Multiplicative Weight Updating

Sanjeew Arora's paper
pmf $x = (x(1), ..., x(n))$ $x \geq 0, \sum x_i = 1$ where x is unknown to the learner
At time t, adversary chooses $f_t \in 0, 1^n$, calculates $< f_t, x >= sum_{i=1}^n f_t(i)x_i \in [0, 1]$ releases to the learner
Learning alg: parameter δ, ϵ
Initialize $x_0 = (1/n, ..., 1/n)$ For $t = 1, ..., T$ if $(< f_t, x > - < f_t, x_t >) > \delta$
$x_t[i] \leftarrow x_{t-1}[i](1 + \epsilon)$ for all i s.t. $f_t(i) = 1$ else keeps the same. Then normalize x_t. Else if $(< f_t, x > - < f_t, x_t >) < -\delta$ symmetric operation
Thm: For every choice of $f_1, f_2, ...$ the alg goes into these two situations for at most $\frac{2 \log n}{\epsilon\delta}$ (if $0 < \epsilon < \delta$)
(Homework 1 to prove it with potential function)

2 K-Cluster

Criterion: k-cluster, learn $c_1, ...c_k$ (cluster center) data: $x_1, ..., x_n \in R^d$ and $x_i \xrightarrow{nearest} c_{j(i)}$
Objective: $\min\limits_{c_1,,,c_k} \sum_{i=1}^n ||x_i - c_{j(i)}||^2$
① Partition
② Compute $c_1, ..., c_k$
③ Partition $x_i \mapsto c_{j(i)}$
Which is k-means
k-means++
Random Initialization: $c_1, ..., c_k$
$\phi_{OPT} = \sum_i ||x_i - c_{j(i)}||^2$
$\phi_{K-means++}$ =??? s.t. $E[\phi_{K-means++}] \leq 8(\log k + 2)\phi_{OPT}$
$c_i \sim \frac{D(x)^2}{\sum_x D(x)^2}$ where $D(x) = ||x - C_{(x)}||$ and $C_{(x)} = argmin||x - c_j||^2, j < i$

3 Reinforcement Learning

Trial and Error

Markov Decision Process Definition(MDP): S(State), P(Probability), A(Action), R(Reward). But actually it should be written as $P^{S}_{S,A}$, $R_{S,A}$ or Transition prob $P(S_{t+1}|S_t, a_t)$ and Reward $R_{t+1} = E[R(S_t, a_t)]$
Goal: Maximize long-term reward. At every time t reward, $\gamma \in (0, 1]$, then $\max G_t := R_{t+1} + \gamma R_{t+2} + ...$
Policy $S \mapsto a$ and $\pi : S \mapsto A$, Given policy π, value function $v_\pi(s) = E[R_{t+1} + \gamma R_{t+1} + ...]$
Action Value Function: $q_\pi(s, a) := E[R_{t+1} + \gamma R_{t+2} + ...|S_t = s, A_t = a]$ at time t, take action a, then follow π at time $t + 1$ and so on.

Bellman Equation Given policy π, $v_\pi(s) = R(s, \pi(s))+\gamma \sum_{s'} P(s'|s, \pi(s))v_\pi(s')$ then by vector and matrix it is $v_\pi = R^{(\pi)} + \gamma P^{(\pi)} v_\pi$
Now denote $\phi_\pi(v) := R^{(\pi)} + \gamma P^{(\pi)} v$ which is called Bellman Expectation Operator.
Thm For $\gamma \in (0, 1)$, define $d(v, v') = \max s \in S|v(s) - v'(s)|$ then Bellman Expectation Operator is a Contraction mapping. So by iteration we can simply calculate $v_\pi(s)$ by Bellman Operator.
$q_\pi(s, a) = R(s, a)+\gamma \sum_{s'} P(s'|s, a)+q_\pi(s', \pi(s'))$ so it is an evaluation for policy π