CSE599s, Spring 2012, Online Learning Lecture 16 - 05/17/2012

Combinatorial Bandits

Lecturer: Ofer Dekel *Scribe: Sergey Feldman*

1 Review - Bandit Convex Optimization

First, let's review the bandit convex optimization problem.

1. The adversary chooses $f_1, \ldots, f_T$, where each $f_t : \mathcal{W} \to \mathbb{R}$.
2. For $t = 1, \ldots, T$ the player
   - chooses $w_t \in \mathcal{W}$ (in a randomized fashion), and
   - suffers loss $f_t(w_t)$, observing f_t *only* at this point.

2 Online Shortest Path Problem

We're going to study *bandits with combinatorial action sets* (aka combinatorial bandits) by looking at one specific example: the online shortest path game. First, some graph notation.

- Let the graph $G = (V, E)$ be composed of a set of vertices V that are connected by a set of directed edges $E \subseteq V^2$.
- Let $u \in V$ and $v \in V$ be the *source* vertex and the *sink* vertex, respectively.
- A (simple) path from u to v is a set of edges that leads from u to v, where "simple" indicates that each edge appears only once and the path is cycle free.
- Let the set of all simple paths from u to v be written as $\text{paths}(u, v)$.

The game is played as follows.

1. The adversary chooses $f_1, \ldots, f_T$, where each $f_t : E \to [0, 1]$.
2. For $t = 1, \ldots, T$ the player
   - plays path $p_t \in \text{paths}(u, v)$, and
   - suffers loss $f_t(p_t) = \sum_{e \in p_t} f_t(e)$, observing only the total loss $f_t(p_t)$ and *not* any of the individual terms $f_t(e)$.

The goal of the online shortest path game is to minimize the expected regret:

$$E\left[\sum_{t=1}^{T} f_t(p_t)\right] - \min_{p \in \text{paths}(u,v)} \sum_{t=1}^{T} f_t(p),$$

where the expectation is taken over the player's internal randomization. Some fruitful observations:

- This game looks very similar to the k-armed bandit problem, where each arm represents a path in $\text{paths}(u, v)$.

- The loss of each arm is bounded:
$$\max_{p \in \text{paths}(u,v)} |p| \leq |E|.$$

However, there is a hefty obstacle – $|\text{paths}(u,v)|$ may be *exponential* in $|E|$, which will lead to (a) exponential regret (something like $\sqrt{T2^{|E|}}$) and (b) exponential complexity.

They key to solving this problem is exploiting its structure. Note that while there may be an exponential number of paths, they are all composed of the same (relatively) small number of edges. We will make this problem tractable in three steps:

1. Reformulate the problem using "integer (binary) linear program" constraints.
2. Perform a convex relaxation.
3. Perform derandomization-randomization.

Step 1 - Problem Reformulation

First, let's enumerate the edges and redefine $p_t \in \{0,1\}^{|E|}$ to be a vector where each coordinate corresponds to the absence or presence of an edge in the path p_t. In other words:

$$\text{for } e \in \{1, \ldots, |E|\} \; p_{t,e} = 1 \text{ iff the corresponding edge is in the path.}$$

Let $g_t = (f_t(1), \ldots, f_t(|E|))$. Then we have that the loss of a path is linear:

$$f_t(p_t) = g_t \cdot p_t = \sum_{e \in E} p_{t,e} g_{t,e}.$$

The following set of constraints on p_t is equivalent to a graph theoretic definition of a simple path from u to v (this is a lemma, but we omit the proof).

- The path p_t is cycle free.
- The path starts at u:
$$\sum_{e=(u,*)} p_{t,e} = 1.$$
- The path ends at v:
$$\sum_{e=(*,v)} p_{t,e} = 1.$$
- Flow is conserved:
$$\forall z \notin \{u,v\}, \sum_{e=(*,z)} p_{t,e} = \sum_{e=(z,*)} p_{t,e}.$$

Step 2 - Convex Relaxation

We're going to pretend that the player can play $w_t \in \textbf{conv}(\text{paths}(u,v))$, i.e. the player can play a point in the convex hull of the paths. The convex hull is the set of all convex combinations of paths in $\{0,1\}^{|E|}$, which is a subset of the hypercube $[0,1]^{|E|}$. Mathematically, $\textbf{conv}(\text{paths}(u,v)) =$

- $\forall e \; w_{t,e} \geq 0, \; w_{t,e} \leq 1.$
- $\sum_{e=(u,*)} w_{t,e} = 1.$

- $\sum_{e=(*,v)} w_{t,e} = 1$.
- $\forall z \notin \{u, v\}$, $\sum_{e=(*,z)} w_{t,e} = \sum_{e=(z,*)} w_{t,e}$.

The loss is still linear

$$f_t(w_t) = g_t \cdot w_t,$$

and now there is a *convex* set of actions, which means we can use the bandit convex optimization framework! For linear loss functions, we can achieve $O(\sqrt{T})$ loss.

Step 3 - Derandomization/Randomization

Using the bandit convex optimization algorithm, we have w_t at each round. But we need to play an actual path p_t. To do so we're going to add *another* layer of randomness. To see how, first note that $w_t \in \mathbf{conv}(\text{paths}(u, v))$ means:

$$\exists\ p_1, \ldots, p_s \in \text{paths}(u, v) \text{ and } \alpha_1, \ldots, \alpha_s \in \mathbb{R}$$

$$\text{s.t. } \alpha_i \geq 0,\ \sum_{i=1}^{s} \alpha_i, \text{ and } \sum_{i=1}^{s} \alpha_i p_i = w_t.$$

In other words, any point in the convex hull of paths can be written as a convex sum of a subset of the paths. What we will do is find a set of $\{\alpha_i\}$ and play path p_i with probability α_i. This is going to work out because of unbiasedness:

$$E[p_i | w_t] = w_t,$$

and, since the loss is linear,

$$E[f_t(p_t)] = E[f_t(w_t)] = g_t \cdot w_t,$$

which will ensure that our regret bound holds. So now we have to make sure that we can find the weights $\{\alpha_i\}$ in polynomial time, or we have just moved the exponential part of the algorithm around, achieving nothing.

Theorem 1. *For $\sum_{i=1}^{s} \alpha_i p_i$, we can find $s \leq |E|$ and $p_i, \ldots, p_s$ and $\alpha_1, \ldots, \alpha_s$ in $O(|E|^2)$ time.*

Proof. A proof by construction - the following is an algorithm for finding $p_i, \ldots, p_s$ and $\alpha_1, \ldots, \alpha_s$. We define $w = w_t$ for simplicity of notation.

For $i = 1, \ldots, |E|$:

1. Find an "augmenting path" p_i in the graph G where the edges are weighted by w. (All edges in an augmenting path have strictly positive weights.)
2. Define $\alpha_i = \min_{e \in p_i} w_e$.
3. Update $w \leftarrow w - \alpha_i p_i$.

We can use a *depth first* search to find an augmenting path in $O(|E|)$ time (if it exists - see subsequent Lemma 1). And there are at most $O(|E|)$ steps in the outer for-loop (see subsequent Lemma 2). Therefore, the total complexity is $O(|E|^2)$. □ □

We now need a few lemmas to round out the proof.

- Lemma 1: $\exists$ an augmenting path until $w = (0, \ldots, 0)$. Proof sketch: flow conservation always holds at each step.
- Lemma 2: It takes $|E|$ iterations to get to $w = (0, \ldots, 0)$. Proof sketch: on each iteration, exactly one edge becomes zero.

The algorithm outlined above is basically the Caratheodory Theorem (see Wikipedia):

Theorem 2. *If $S \subseteq \mathbb{R}^n$ and $x \in \textbf{\textit{conv}}(S)$, then $\exists\ s_1, \ldots, s_{n+1} \in S$ s.t. $x \in \textbf{\textit{conv}}(\{s_1, \ldots, s_{n+1}\})$.*

3 Online Bipartite Matching

Consider the problem of online dating, in which we'd like to match k men and k women to each other on every round - online bipartite matching. The cardinality of the action space is huge: $k!$, but here too we can use the structure of the problem to get good regret bounds.

Let $g_t \in [0,1]^{k \times k}$ be a $k \times k$ loss matrix, where

$$g_t(i,j) = 0 \Leftrightarrow i \text{ and } j \text{ didn't like each other.}$$

The goal is then to minimize

$$\mathbf{1}^T g_t \mathbf{1}.$$