Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient

Abraham D. Flaxman, CMU Math
Adam Tauman Kalai, TTI-Chicago
H. Brendan McMahan, CMU CS

May 25, 2006

Outline

Online Analysis
- Example: Multi-Armed Bandit
- General Setting: Online Convex Optimization

Results
- Bandit Gradient Descent: Algorithm
- Bandit Gradient Descent: Analysis

Example: Multi-Armed Bandit Problem

Pittsburgh plans casino with 1000 slot machines.

Example: Multi-Armed Bandit Problem

One (of many) problems this raises:

- Which of the 1000 machines should you play?

Example: Multi-Armed Bandit Problem

One (of many) problems this raises:

- Which of the 1000 machines should you play?

A traditional approach:

- Assume that each machine i pays out independently with probability p_i
- Develop strategy accordingly

Example: Multi-Armed Bandit Problem

One (of many) problems this raises:

- Which of the 1000 machines should you play?

A traditional approach:

- Assume that each machine i pays out independently with probability p_i
- Develop strategy accordingly

Would be nice not to assume that the machines are so well behaved.

Example: Adversarial Multi-Armed Bandit

The CS-Theory approach to removing the assumption:

- An adversary controls when the machines pay out

Example: Adversarial Multi-Armed Bandit

The CS-Theory approach to removing the assumption:

- An adversary controls when the machines pay out
- The adversary knows everything about your algorithm, *except* the results of random coin tosses

Example: Adversarial Multi-Armed Bandit

The CS-Theory approach to removing the assumption:

- An adversary controls when the machines pay out
- The adversary knows everything about your algorithm, *except* the results of random coin tosses
- How do you know if you are doing well?

Competitive Analysis and Regret

To see how well you've done:

- Compare with the best you could have done if you knew the future

Competitive Analysis and Regret

To see how well you've done:

- Compare with the best you could have done if you knew the future

- competitive ratio $:= z_{\text{offline}}/z_{\text{online}}$
- regret $:= z_{\text{offline}} - z_{\text{online}}$

Competitive Analysis and Regret

To see how well you've done:

- Compare with the best you could have done if you knew the future

- competitive ratio $:= z_{\text{offline}}/z_{\text{online}}$
- regret $:= z_{\text{offline}} - z_{\text{online}}$

What is z_{offline}?

Competitive Analysis and Regret

To see how well you've done:

- Compare with the best you could have done if you knew the future
- competitive ratio $:= z_{\text{offline}} / z_{\text{online}}$
- regret $:= z_{\text{offline}} - z_{\text{online}}$

What is z_{offline}?

- Be nice if it was best sequence of machines to play

Competitive Analysis and Regret

To see how well you've done:

- Compare with the best you could have done if you knew the future

- competitive ratio $:= z_{\text{offline}}/z_{\text{online}}$
- regret $:= z_{\text{offline}} - z_{\text{online}}$

What is z_{offline}?

- Be nice if it was best sequence of machines to play
- To have results, make it best *single* machine to play

Learning from Expert Advice

- Many other problems also fit into this framework

Learning from Expert Advice

- Many other problems also fit into this framework
- For example, learning from expert advice

Learning from Expert Advice

- Many other problems also fit into this framework
- For example, learning from expert advice
- But for computational reasons, a more general setting can be convenient

General Setting: Online Convex Optimization

General Setting: Online Convex Optimization

- Bounded convex set $S \subseteq \mathbb{R}^d$, constant $C > 0$

General Setting: Online Convex Optimization

- Bounded convex set $S \subseteq \mathbb{R}^d$, constant $C > 0$
- At time t, simultaneously,
  - Adversary chooses convex function $c_t : S \to [-C, C]$
  - We choose point $x_t \in S$

General Setting: Online Convex Optimization

- Bounded convex set $S \subseteq \mathbb{R}^d$, constant $C > 0$
- At time t, simultaneously,
  - Adversary chooses convex function $c_t \colon S \to [-C, C]$
  - We choose point $x_t \in S$
- We pay adversary $c_t(x_t)$

General Setting: Online Convex Optimization

- Bounded convex set $S \subseteq \mathbb{R}^d$, constant $C > 0$
- At time t, simultaneously,
  - Adversary chooses convex function $c_t : S \to [-C, C]$
  - We choose point $x_t \in S$
- We pay adversary $c_t(x_t)$

Goal: choose a sequence of x_t to make $\sum_t c_t(x_t)$ small

General Setting: Online Convex Optimization

- Bounded convex set $S \subseteq \mathbb{R}^d$, constant $C > 0$
- At time t, simultaneously,
  - Adversary chooses convex function $c_t : S \to [-C, C]$
  - We choose point $x_t \in S$
- We pay adversary $c_t(x_t)$

Goal: choose a sequence of x_t to make $\sum_t c_t(x_t)$ small

How we tell if we've done well: if *regret* is small

$$\text{regret} := \min_{x \in S} \left\{ \sum_t c_t(x) \right\} - \sum_t c_t(x_t)$$

An upper bound on regret

An upper bound on regret

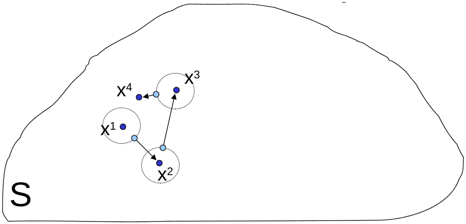
Theorem

There is a randomized algorithm which produces a sequence of x_t*, so that if* $S \subseteq \mathbb{R}^d$ *and each* c_t *takes values in* $[-C, C]$ *then*

$$\mathbb{E}[\mathit{regret}] = \mathbb{E}\left[\min_{x \in S}\left\{\sum_{t=1}^{n} c_t(x)\right\} - \sum_{t=1}^{n} c_t(x_t)\right] \leq 6Cdn^{5/6}.$$

The Randomized Algorithm

Algorithm is a form of gradient descent



Main trick: a random variable approximating the gradient and formed by a single evaluation of the function

Analysis of algorithm

The flavor of the analysis is this:

Analysis of algorithm

The flavor of the analysis is this:

$$\|x_t - x_\star\|^2/2\eta = \|\mathbf{P}(x_t - \eta \cdot g_t) - x_\star\|^2 = \ldots$$

Analysis of algorithm

The flavor of the analysis is this:

$$\|x_t - x_\star\|^2/2\eta = \|\mathbf{P}(x_t - \eta \cdot g_t) - x_\star\|^2 = \ldots$$

$$c_t(x_t) - c_t(x_\star) \leq \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2}{2\eta} + \frac{\eta G^2}{2}.$$

Analysis of algorithm

The flavor of the analysis is this:

$$\|x_t - x_\star\|^2/2\eta = \|\mathbf{P}(x_t - \eta \cdot g_t) - x_\star\|^2 = \dots$$

$$c_t(x_t) - c_t(x_\star) \le \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2}{2\eta} + \frac{\eta G^2}{2}.$$

$$\text{regret} = \sum_{t=1}^{n} c_t(x_t) - c_t(x_\star) \le \frac{\|x_1 - x_\star\|^2}{2\eta} + n\frac{\eta G^2}{2}.$$

Analysis of algorithm

The flavor of the analysis is this:

$$\|x_t - x_\star\|^2/2\eta = \|\mathbf{P}(x_t - \eta \cdot g_t) - x_\star\|^2 = \dots$$

$$c_t(x_t) - c_t(x_\star) \leq \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2}{2\eta} + \frac{\eta G^2}{2}.$$

$$\text{regret} = \sum_{t=1}^{n} c_t(x_t) - c_t(x_\star) \leq \frac{\|x_1 - x_\star\|^2}{2\eta} + n\frac{\eta G^2}{2}.$$

Take $\eta = 1/\sqrt{n}$.

Analysis of algorithm

The flavor of the analysis is this:

$$\|x_t - x_\star\|^2/2\eta = \|\mathbf{P}(x_t - \eta \cdot g_t) - x_\star\|^2 = \ldots$$

$$c_t(x_t) - c_t(x_\star) \leq \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2}{2\eta} + \frac{\eta G^2}{2}.$$

$$\text{regret} = \sum_{t=1}^{n} c_t(x_t) - c_t(x_\star) \leq \frac{\|x_1 - x_\star\|^2}{2\eta} + n\frac{\eta G^2}{2}.$$

Take $\eta = 1/\sqrt{n}$.

- Full Details:
  - A. Flaxman, A. Kalai, H. McMahon, Online convex optimization in the bandit setting: gradient descent without a gradient, Symposium of Discrete Algorithms (SODA), 2005.

Conclusion

Conclusion

- Online convex optimization in the bandit setting

Conclusion

- Online convex optimization in the bandit setting
  - Analysis in an adversarial setting

Conclusion

- Online convex optimization in the bandit setting
  - Analysis in an adversarial setting
- Exists algorithm with have regret $\leq 6Cdn^{5/6}$

Conclusion

- Online convex optimization in the bandit setting
  - Analysis in an adversarial setting
- Exists algorithm with have regret $\leq 6Cdn^{5/6}$
- Streaming Algorithms?

Conclusion

- Online convex optimization in the bandit setting
  - Analysis in an adversarial setting
- Exists algorithm with have regret $\leq 6Cdn^{5/6}$
- Streaming Algorithms?
  - ϵ-approximate solution in ϵ^6 passes over the data