CSE 521: Design & Analysis of Algorithms I

Online algorithms

Online algorithms

- Making decisions under uncertainty
  - Many ways to analyze.
    - Competitive analysis: worst case ratio of performance of online algorithm to performance of optimal with hindsight.
    - Regret: worst case difference in performance.

Competitive Analysis Examples

- Ski Rental
- Finding a hole in a fence
- List Update
- (Splay trees)
- Paging and caching

Ski Rental

- Renting skis costs $50
- Buying skis costs $500
- What should you do? Rent or buy?
- If you knew the future…

Finding a hole in a fence

- You're standing in front of a long fence. There is a hole somewhere, but you don't know where it is.
- Each period, you can take a step left or a step right.
- Your goal is to find the hole with the fewest number of steps.

List update

- Maintain a linked list of n items (numbered 1..n)
- Perform a sequence of lookups, each one takes time = position of element in list.
- Rule *: After a lookup, requested item may be put anywhere in the list between the start and its position before the lookup.
- Additional swaps cost 1 each.
- Have to make decisions online.

A good algorithm: Move to Front

- Always move requested item to front of list.
- Theorem: MTF has competitive ratio 2.
- Proof:
  - Rule * → Rule **: After a request required to move the requested element to front.
  - Can simulate an algorithm with cost C under rule * by an algorithm with cost at most 2C under rule **.
  - With Rule **, MTF is optimal.

Online learning and Multiplicative Weights Update Method

- Method has been used in many variants over the years
- From a recent survey by Arora, Hazan, Kale:
  - This "meta algorithm and its analysis are simple and useful enough that they should be viewed as a basic tool taught to all algorithms students together with divide-and-conquer, dynamic programming, random sampling, and the like."
  - http://www.cs.princeton.edu/~satyen/papers/mw-survey.pdf

Online choice from experts

- Simple case: Stock market direction
  - **n** experts
  - every day each expert **i** makes a binary guess/ prediction (up=**+1** or down=**-1**)
  - at end of the day can observe the outcome of what the market did that day
- Goal: Is there a strategy that allows us to do nearly as well as the best of these experts in hindsight?

Simpler question

- **n** experts
- every day each expert **i** makes a binary guess/ prediction (up=**+1** or down=**-1**)
- at end of the day can observe the outcome of what the market did that day
- One of them is **perfect** – never makes a mistake. We just don't know which one.

- Goal: Can we find a strategy that makes no more than lg(n) mistakes?

What if no expert is perfect?

- **Intuition:** making a mistake doesn't completely disquality an expert. So, instead of crossing off, just lower its weight.
- **Weighted majority algorithm:**
  - **Start with all experts having weight 1.**
  - **Predict based on majority vote.**
  - **Penalize mistakes by cutting weight in half.**
- **Claim:** do nearly as well as best expert in hindsight.
- Can use this to combine multiple algs to do nearly as well as best in hindsight.