

Homework due 11/19. Please write the name of any collaborator you worked with.

1. The offline  $k$ -server problem. Given a metric space with set  $X$  and distance  $d$ . Given a sequence of accesses,  $\sigma_1, \sigma_2, \dots, \sigma_T$ , describe an offline dynamic-programming algorithm to compute the minimum cost way of servicing those requests with  $k$  servers. The runtime of your algorithm should be polynomial in  $n^k$  and  $T$ , where  $n = |X|$  is the number of points in the set.
2. The offline  $k$ -server problem using flows. Give an algorithm for the above problem that uses min-cost maximum flow. In a min-cost maximum flow problem<sup>1</sup>, the input is a directed graph. For each edge there is a non-negative capacity and non-negative cost. A min-cost max-flow on the graph is a flow of minimal cost among all flows that have maximum capacity. You can assume that we have an efficient algorithm for this problem. You can also assume that, if all capacities and costs are integer, it always returns an integer flow, i.e., a union of paths. Show how to reduce the offline  $k$ -server problem to a min-cost max-flow problem on a graph with  $O(nT)$  nodes and  $O(n^2T)$  edges.
3. Frequency-count algorithm for Universal Portfolios.
  - (a) For any given sequence of price relatives, show that computing the optimal constant rebalanced portfolio (CRP) in hindsight is possible efficiently. In particular, given a sequence of length  $T$  on price relatives of  $n$  stocks,  $r^1, r^2, \dots, r^T \in \mathbb{R}^n$ , the return of a particular CRP  $x \in \Delta = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1 \wedge x_i \geq 0\}$  is

$$\prod_{t=1}^T x \cdot r^t.$$

Show that the above is a concave function of  $x$ . Oops, Ivona pointed out that it is not concave. What is concave is the log of

---

<sup>1</sup>If you are not familiar with these problems, you can find them in most algorithms books, for example “Introduction to Algorithms,” page 787, by Cormen Leiserson and Rivest.

the above expression. Since maximizing  $\log f(x)$  is equivalent to maximizing  $f(x)$ , it suffices to maximize  $\log f(x)$ . In particular, you need to show that  $f(x) = \log \left( \prod_{t=1}^T x \cdot r^t \right)$  is concave in  $x$ . Thus, we can use existing tools for efficiently maximizing a concave function over a convex set.

- (b) A natural algorithm for this problem would be like the earlier frequency count algorithms: on period  $t$ , use

$$\arg \max_{x \in \Delta} \prod_{j=1}^{t-1} x \cdot r^j,$$

i.e., the best single portfolio on the first  $t - 1$  periods. From part (a), we know that it is possible to compute this efficiently, in time polynomial in the number of stocks and number of periods. Is it possible to implement this algorithm by keeping track of  $n$  “frequencies,” i.e.,  $n$  totals which can be easily updated and used to compute the best portfolio in hindsight? If so, what are these frequencies?

- (c) Show that the frequency count algorithm described above performs poorly. Give a short sequence on which it performs much worse than the best algorithm in hindsight. Make the ratio between their performance as poor as possible.
- (d) (Harder: you get major extra credit if you solve this one.) Consider the algorithm that, on period  $t$ , uses

$$\arg \max_{x \in \Delta} \prod_{i=1}^n x_i \prod_{j=1}^{t-1} x \cdot r^j$$

This is equivalent to doing frequency count with  $n$  pretend periods that we added at the beginning. During the  $i$ th pretend period, every stock crashed (price relative of 0, or if you prefer not to think about zero, you can choose an arbitrarily small  $\epsilon$ ) except the  $i$ th stock, which remained constant. In other words, the price relatives were  $(0, \dots, 0, 1, 0, \dots, 0)$  with a 1 in the  $i$ th position. (Notice that keeping all your money in any single stock will result in a loss of all of your money.)

Analyze the per-period competitive ratio of this algorithm. Is it *universal*, i.e., does its per-period competitive ratio approach 1 as  $T$  approaches infinity?