Spectral Graph Theory and Algorithms, Summer 2017 29 June 2017

Problem Set 8

Due Date: 30 June 2017

1. We can use the multiplicative weights algorithm to upper bound the costs we pay per time step rather than to lower bound the value we get per time step. Suppose that in time step t, if we make decision j, we pay a cost $c_t(j) \in [-1, 1]$. We will want to modify the algorithm so that if the cost of a decision is high, we are less likely to make that decision, so instead of multiplying each weight $w_t(j)$ by $(1 + \epsilon v_t(j))$ we multiply it by $(1 - \epsilon c_t(j))$.

   Show that if $\epsilon \leq 1/2$, then after T rounds, for any decision j, we have that the expected cost of our solution, $\sum_{t=1}^{T} \sum_{i=1}^{N} c_t(i) \cdot p_t(i)$, is at most

   $$\sum_{t=1}^{T} c_t(j) + \epsilon \sum_{t=1}^{T} |c_t(j)| + \frac{1}{\epsilon} \ln N.$$

   It may be help to have the following inequalities:

   - $1 - x \leq e^{-x}$ for any x;
   - $(1 - \epsilon x) \geq (1 - \epsilon)^x$ for $x \in [0, 1]$;
   - $(1 - \epsilon x) \geq (1 + \epsilon)^{-x}$ for $x \in [-1, 0]$;
   - $\ln(1/(1 - \epsilon)) \leq \epsilon + \epsilon^2$ when $0 < \epsilon \leq 1/2$;
   - $\ln(1 + \epsilon) \geq \epsilon - \epsilon^2$ when $0 < \epsilon \leq 1/2$.