

Problem Set 4

Due Date: November 22, 2016

As a reminder, the collaboration policy from the syllabus is as follows:

Your work on problem sets and exams should be your own. You may discuss approaches to problems with other students, but as a general guideline, such discussions may not involve taking notes. You must write up solutions on your own independently, and acknowledge anyone with whom you discussed the problem by writing their names on your problem set. You may not use papers or books or other sources (e.g. material from the web) to help obtain your solution.

Problem sets are due in class at the start of the lecture. If you would like an extension, please arrange it in advance.

1. We can use the (standard scalar) 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 $c_t(j) \in [-1, 1]$. 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

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

2. As with the multiplicative weights question from above, we can think about the matrix multiplicative weight algorithm as minimizing costs, rather than maximizing value. Suppose that in each time step t , we must choose a unit norm vector u with $\|u\| = 1$, and we must pay a cost $u^T M_t u$ for matrix M_t revealed at that time step with $0 \preceq M_t \preceq I$. Consider the variant of the matrix multiplicative weight algorithm given below.

- (a) As usual, we will compare the expected cost of the algorithm with the cost of the best possible fixed choice (that is, cost of a fixed vector u , $\|u\| = 1$). Show that the expected cost of the algorithm is

$$\sum_{t=1}^T P_t \bullet M_t,$$

while the cost of the best possible fixed choice is

$$\lambda_{\min} \left(\sum_{t=1}^T M_t \right).$$

Algorithm 1: Matrix Multiplicative Weights

$W_1 \leftarrow I$
for $t \leftarrow 1$ **to** T **do**
 $P_t \leftarrow W_t / \text{tr}(W_t)$
 Let λ_{it}, x_{it} be eigenvalues, eigenvectors of P_t , x_{it} orthonormal
 Choose x_{it} with probability λ_{it} , pay cost $x_{it}^T M_t x_{it}$.
 $W_{t+1} \leftarrow \exp(-\epsilon \sum_{k=1}^t M_k)$
end

(b) Prove that if A is a matrix with $0 \preceq A \preceq I$, then

$$\exp(-\epsilon A) \preceq I - (1 - e^{-\epsilon})A.$$

(c) Prove that for $0 \leq \epsilon \leq 1/2$, the expected cost of the algorithm is not much more than the cost of the best possible fixed choice by showing that

$$\sum_{t=1}^T P_t \bullet M_t \leq \frac{1}{1-\epsilon} \lambda_{\min} \left(\sum_{t=1}^T M_t \right) + \frac{1}{\epsilon(1-\epsilon)} \ln n.$$

It may help to know that for $0 \leq \epsilon \leq 1/2$, $1 - e^{-\epsilon} \geq \epsilon(1 - \epsilon)$.

3. Prove the Sherman-Morrison formula, namely for X nonsingular and symmetric and a vector v ,

$$(X - vv^T)^{-1} = X^{-1} + \frac{X^{-1}vv^TX^{-1}}{1 - v^TX^{-1}v}.$$

4. We say that a vector $x \in \mathbb{R}^{|E|}$ is *in the spanning tree polytope* if $\sum_{(i,j) \in E} x(i,j) = n - 1$, and for any set $S \subset V$, $\sum_{(i,j) \in E(S)} x(i,j) \leq |S| - 1$ (recall that $E(S)$ is the set of all edges with both endpoints in S). If x is in the spanning tree polytope, then it can be written as a convex combination of spanning trees; that is, if \mathcal{T} is the set of all spanning trees in a graph G , and for spanning tree T , $\chi_T \in \{0,1\}^{|E|}$ is the characteristic vector of the tree (that is, $\chi_T(i,j) = 1$ if $(i,j) \in T$ and is 0 otherwise), then we can write

$$x = \sum_{T \in \mathcal{T}} \alpha_T \chi_T$$

for some $\alpha_T \geq 0$ such that $\sum_{T \in \mathcal{T}} \alpha_T = 1$.

Given a point x in the spanning tree polytope, let L_x be the Laplacian associated with the weights x , so that

$$L_x = \sum_{(i,j) \in E} x(i,j)(e_i - e_j)(e_i - e_j)^T.$$

In this problem we will show that it is possible to use some of the machinery of the linear-sized sparsifier proof to pick a set F of $n/2$ edges such that

$$\sum_{(i,j) \in F} (e_i - e_j)(e_i - e_j)^T \preceq 35 L_x.$$

F will be a set, not a multiset. In what follows we assume that $n \geq 3$. We will use the algorithm below.

Algorithm 2: FindForest

```

 $A \leftarrow 0$ 
 $F \leftarrow \emptyset$ 
 $u \leftarrow u_0 \leftarrow 20$ 
 $\Delta u \leftarrow 20/(n-1)$ 
 $U(u, A) \equiv \text{tr}((uI - A)^{-1})$ 
while  $|F| < n/2$  do
    Pick  $(i, j) \in E - F$  such that  $F \cup \{(i, j)\}$  is acyclic
     $F \leftarrow F \cup \{(i, j)\}$ 
     $A \leftarrow A + z_{(i,j)} z_{(i,j)}^T$ 
     $u \leftarrow u + \Delta u$ 
end
return  $F$ 

```

- (a) Show that there exist vectors $z_{(i,j)}$ with $\sum_{(i,j) \in E} x(i, j) z_{(i,j)} z_{(i,j)}^T = I$ (up to our usual fudging about I) such that

$$\sum_{(i,j) \in F} (e_i - e_j)(e_i - e_j)^T \preceq 35 L_x$$

if and only if

$$\lambda_{\max} \left(\sum_{(i,j) \in F} z_{(i,j)} z_{(i,j)}^T \right) \leq 35.$$

- (b) Suppose we run the algorithm FindForest shown; note that it is essentially the algorithm for finding a linear-sized sparsifier, but without the lower bounds. In each iteration we must pick an edge e to add to F such that $e \notin F$, and adding e to F does not make F cyclic. We want to show that such an edge must always exist as long as $|F| < n/2$.

Assume for the moment that we pick such an edge by sampling edge (i, j) with probability $x(i, j)/(n-1)$. Show that with probability at least $7/8$, $U(u + \Delta u, A + z_{(k,\ell)} z_{(k,\ell)}^T) \leq U(u, A)$ if we pick edge (k, ℓ) via our sampling.

- (c) Show that the probability we pick an edge already in F or an edge that closes a cycle is at most $3/4$.
- (d) Conclude that it is possible in each iteration to pick an edge e to add to F such that $e \notin F$, and adding e to F does not make F cyclic, and such that $U(u + \Delta u, A + z_e z_e^T) \leq U(u, A)$.
- (e) Finish the proof by arguing that the chosen set of edges in F must be such that

$$\sum_{(i,j) \in F} (e_i - e_j)(e_i - e_j)^T \preceq 35 L_x.$$