PRINCETON UNIVERSITY FALL '16 COS 521: ADVANCED ALGORITHMS

# Homework 2

Out: *Oct 12* Due: *Oct 21*

*You can collaborate with your classmates, but be sure to list your collaborators with your answer. If you get help from a published source (book, paper etc.), cite that. The answer must be written by you and you should not be looking at any other source while writing it. Also, limit your answers to one page, preferably less —you just need to give enough detail to convince the grader.*

*Typeset your answer in latex (if you don't know latex, scan your handwritten work into pdf form before submitting). To simplify grading, submit answers in the numbered order listed below, and also make sure your name appears on every page.*

§1 (Approximate LP Solving via Multiplicative Weights) This exercise develops an algorithm to approximately solve Linear Programs.

Consider the problem of finding if a system of linear inequalities as below admits a solution - i.e., whether the system is feasible. This is an example of a feasibility linear program and while it appears restrictive, one can use it solve arbitrary linear programs to obtain approximate solutions.

$$\begin{aligned} a_1^\top x &\geq b_1 \\ a_2^\top x &\geq b_2 \\ &\vdots \\ a_m^\top x &\geq b_m \\ x_i &\geq 0 \ \forall\ i \in [n] \\ \sum_{i=1}^n x_i &= 1. \end{aligned} \tag{1}$$

1) (Duality) Show a simple method to solve the following linear program with two non-trivial constraints for any weights $w_1, w_2, \dots, w_m$.

$$\begin{aligned} \max \sum_{j=1}^m & w_j (a_j^\top x - b_j) \\ x_i &\geq 0 \ \forall i \in [n] \\ \sum_{i=1}^n x_i &= 1. \end{aligned} \tag{2}$$

Conclude that if there are non-negative weights $w_1, w_2, \dots, w_m$ such that the value of the program above is negative, then the system (1) is infeasible (this LP is said to be the *dual* of the LP in (1).)

2) The above setting of finding weights that certify infeasibility of (1) might remind you of the setting of weighting the experts via multiplicative weights update rule discussed in the class. Use these ideas to obtain an algorithm that a) either finds a set of non-negative weights certifying infeasibility of LP in (1) or b) finds a solution $x$ that approximately satisfies all the constraints in (1), i.e., for each $1 \leq i \leq m$, $a_j^\top x - b_j \geq -\epsilon$, $x_i \geq 0$ and $\sum_{i=1}^n x_i = 1$. Give a bound, as tight as possible, for the number of update steps required in order to reach the above goal and use it to obtain a running time bound for approximate LP solving. You may use the meta-theorem MW from the lecture as a blackbox.

(Hint: Identify $m$ "experts" - one for each inequality constraint in (1) and maintain a weighting of experts (starting with the uniform weighting of all 1s, say) for times $t = 0, 1, \ldots,$ - these are your progressively improving guesses for the weights. Solve (2) using the weights at time $t$. If the value of (2) is negative, you are done, otherwise think of the "cost" of the $j^{th}$ expert as $a_j^\top x^{(t)} - b_j$ where $x^{(t)}$ is the solution to the LP (2) at time $t$ and update the weights.)

§2 In $\ell_2$ regression you are given datapoints $x_1, x_2, \ldots, x_n \in \Re^k$ and some values $y_1, y_2, \ldots, y_n \in \Re$ and wish to find the "best" linear function that fits this dataset. A frequent choice for best fit is the one with *least squared error*, i.e. find $a \in \Re^k$ that minimizes

$$\sum_{i=1}^{h} |y - a \cdot x_i|^2.$$

Show how to solve this problem in polynomial time (hint: reduce to solving linear equations; at some point you may need a certain matrix to be invertible, which you can assume.).

§3 (Firehouse location) Suppose we model a city as an $m$-point finite metric space with $d(x, y)$ denoting the distance between points $x, y$. These $\binom{m}{2}$ distances (which satisfy triangle inequality) are given as part of the input. The city has $n$ houses located at points $v_1, v_2, \ldots, v_n$ in this metric space. The city wishes to build $k$ firehouses and asks you to help find the best locations $c_1, c_2, \ldots, c_k$ for them, which can be located at any of the $m$ points in the city. The *happiness* of a town resident with the final locations depends upon his distance from the closest firehouse. So you decide to minimize the cost function $\sum_{i=1}^n d(v_i, u_i)$ where $u_i \in \{c_1, c_2, \ldots, c_k\}$ is the firehouse closest to $v_i$. Describe an LP-based algorithm that runs in *poly*$(m)$ time and solves this problem approximately. If OPT is the optimum cost of a solution with $k$ firehouses, your solution is allowed to use $O(k \log m)$ firehouses and have cost at most $(1 + \epsilon)$OPT.

§4 In class we designed a 3/4-approximation for MAX-2SAT using LP rounding. Extend it to a 3/4-approximation for MAX-SAT (i.e., where clauses can have 1 or more variables). Hint: you may also need the following idea: if a clause has size $k$ and we randomly assign values to the variables (i.e., 0/1 with equal probability) then the probability we satisfy it is $1 - 1/2^k$.

§5 You are given data containing grades in different courses for 5 students. As discussed in Lecture 5, we are trying to "explain" the grades as a linear function of the student's aptitude, the easiness of the course and some error term. Denoting by $\text{Grade}_{ij}$ the grade of student $i$ in course $j$ this linear model hypothesizes that

$$\text{Grade}_{ij} = \text{aptitude}_i + \text{easiness}_j + \epsilon_{ij},$$

where $\epsilon_{ij}$ is an error term.

As we saw in class, the problem of finding the best model that minimizes the sum of the $|\epsilon_{ij}|$'s can be solved by an LP. Your goal is to use any standard package for linear programming (Matlab/CVX, Freemat, Sci-Python, Excel etc.; we recommend CVX on matlab) to fit the best model to this data. Include a printout of your code, and the calculated easiness values of all the courses and the aptitudes of all the students.

| | MAT | CHE | ANT | REL | POL | ECO | COS |
|---|---|---|---|---|---|---|---|
| Alex | | | C+ | A | B+ | A- | C+ |
| Billy | B+ | A- | | | A- | B | B |
| Chris | B | B+ | | | A | A- | B+ |
| David | A | | B- | A | | A- | |
| Elise | | B- | C | B+ | B | B | C |

Assume $A = 10, B = 8$ and so on. Let $B+ = 9$ and $A- = 9.5$. (If you use a different numerical conversion please state it clearly.)

§6 (Optimal life partners via MDP) Your friend is trying to find a life partner by going on dates with $n$ people selected for her by an online dating servie. After each date she has two choices: select the latest person she dated and stop the process, or reject this person and continue to date. She has asked you to suggest the optimum *stopping rule*. You can assume that the $n$ persons are all linearly orderable (i.e. given a choice between any two, she is not indifferent and prefers one over the other). The dating service presents the $n$ chosen people in a random order, and her goal is to maximise the chance of ending up with the person that she will like the most among these $n$. (Thus ending up even with her second favorite person out of the $n$ counts as failure; she's a perfectionist.) Represent her actions as an MDP, compute the optimum strategy for her and the expected probability of success by following this strategy.

(Hint: The Optimal rule is of the form: *Date $\gamma n$ people and decide beforehand to pass on them. After that select the first person who is preferable to all people seen so far.* You may also need that $\sum_{k=t_1}^{t_2} \frac{1}{k} \approx \ln \frac{t_2}{t_1}$.)

§7 (extra credit) In question 4 try to design an algorithm that uses $k$ firehouses but has cost $O(\text{OPT})$. (Needs a complicated dependent rounding; you can also try other ideas.) Partial credit available for partial progress.