EECS 127/227AT 10/16/2018
L. El Ghaoui, Alex Bayen

Quiz 2

Your Name:

Your SID:

Instructions:

1. The quiz lasts 45 minutes.
2. Notes are *not* allowed, except for a one-page, two sided cheat sheet.
3. Do not open the exam until you are told to do so.

1. **Multi-period planning (building a house).** We consider the problem of scheduling several tasks to build a house. In the context of the planning problem:

   - each task takes a known amount of time to complete;
   - a task may depend on other tasks, and can only be started once those other tasks are complete;
   - tasks may be worked on simultaneously provided they do not depend on each other.

   The dependency constraints for the problem can be expressed in the following dependency graph. We would like to build the house as fast as possible by completing these

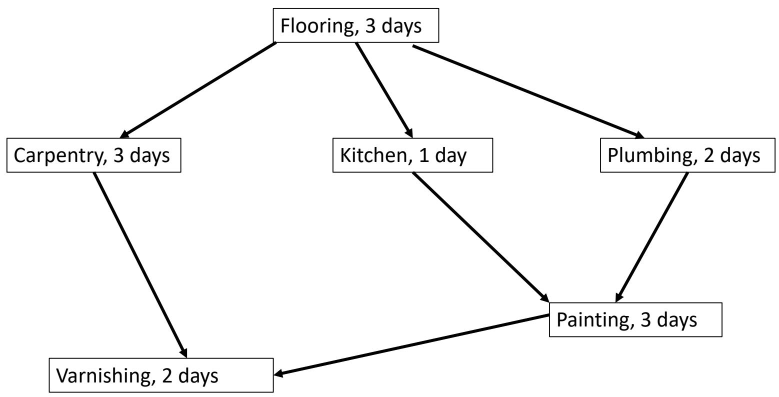

Figure 1: Dependency graph for the construction project we consider.

series of tasks as fast as possible. Each node in the graph represents a tuple containing a task name and the time required to complete it (i.e. the "Flooring" task corresponding to the root node takes 3 days to complete). Moreover, the parents of a given node in the graph correspond to the tasks which must be completed before work on that given task can begin (i.e. we can only start on "Varnishing" after *both* "Carpentry" and "Painting" are completed).

(a) Formulate the problem of minimizing the total time to build the house, subject to the aforementioned constraints in the given dependency graph, as an optimization problem. To do so, let $t_F, t_C, t_K, t_{Pl}, t_{Pa}, t_V$ be the start times of the Flooring, Carpentry, Kitchen, Plumbing, Painting and Varnishing tasks respectively; formulate the problem in terms of these non-negative variables.

(b) Does your formulation of the problem fall within any "nice" problem class (*i.e.* LP/QP/QCQP/SOCP)? If so, state the best class description (i.e. the smallest class containing the problem) and express the problem in "standard" form, clearly defining the relevant matrices/vectors. If not, explain why.

2. **Non-negative matrix factorization.** We consider the problem of matrix factorization, where given a matrix $X \in \mathbb{R}^{n,p}$, we seek a low-rank approximation of the form $X \approx LR^\top$, with $L \in \mathbb{R}^{n,k}$ and $R \in \mathbb{R}^{p,k}$ (usually we pick $k << \min\{n, p\}$). Formally, we will consider solving (variants) of the optimization problem

$$\min_{L,R} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k}, \;\; R \in \mathbb{R}^{p,k}. \tag{1}$$

(a) What is the largest possible rank of the matrix $LR^\top$ referenced in Equation (1)?

(b) Find a solution to the problem in Equation (1). A clear explanation of how to construct the solution to the problem suffices; a mathematical derivation/proof is not necessary.

(c) In many cases of interest, the data matrix X has non-negative entries—for example if the matrix contains ratings of items such as movies. It seems natural then to enforce the additional elementwise constraints $L \geq 0$ and $R \geq 0$ on the matrices L, R. We are thus interested in solving the optimization problem

$$\min_{L,R} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k}, \;\; R \in \mathbb{R}^{p,k}, \;\; L \geq 0, \;\; R \geq 0 \tag{2}$$

You might wonder if the problem in Equation (2) has a unique solution. To gain intuition for this issue, consider the simple case where $n = k = p = 1$. Does this simple problem have a unique solution? Explain why or why not.

(d) Problem (2) is quite difficult to solve in general. In many cases, we will consider *alternating minimization* algorithms on the objective in Equation (2). Such an algorithm moves back and forth between solving for the optimal R and updating R, assuming L is fixed; then solving for the optimal L and updating L, assuming R is fixed. Formally, the subproblem to solve for the optimal L, assuming R is fixed, is

$$\min_{L} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k} \;\; R \in \mathbb{R}^{p,k}, \;\; L \geq 0, \tag{3}$$

(and vice-versa to optimize over R). Can the problem in (3) be formulated in any "nice" problem class (*i.e.* LP/QP/SOCP/ ...)? If so, state the best class description (i.e. the smallest class containing the problem) and justify your answer.

(e) Assuming that that $k = 1$ for simplicity, so that L, R are vectors, compute the gradient[1] of the objective function $\|X - LR^\top\|_F^2$ in Equation (3) with respect to $L \in \mathbb{R}^n$.

[1] This gradient can be useful to derive yet another iterative optimization algorithm to solve Equation (3) itself.