EE 127/227AT 4/10/2018

L. El Ghaoui

Quiz 2: Solutions

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

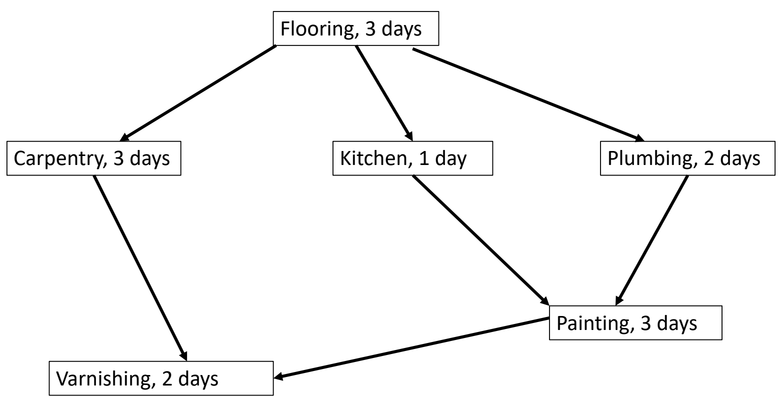

Figure 1: Dependency graph for the construction project we consider.

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.

Solution 1

(a) The carpentry, kitchen, and plumbing tasks cannot begin until the flooring task has finished, and the flooring task takes 3 days to finish. So the first set of constraints are,

$$t_F + 3 \leq t_C, \quad t_F + 3 \leq t_K, \quad t_F + 3 \leq t_{Pl} \tag{1}$$

Following a similar line of reasoning for the other dependencies we have that,

$$t_K + 1 \leq t_{Pa}, t_{Pl} + 2 \leq t_{Pa} \tag{2}$$

and finally,

$$t_C + 3 \leq t_V, t_{Pa} + 3 \leq t_V \tag{3}$$

These are all the constraints implied by the dependency graph. Moreover, in order to enforce non-negativity of the times we simply set $t_F \geq 0$.

The total duration of the house-building process is the difference between the start time of the varnishing task and flooring task plus the time it takes to complete the varnishing task (2 days). So the objective to minimize for the problem is,

$$t_V - t_F + 2 \tag{4}$$

The set of optimization variables (to optimize over) is all of the t variables.

Note that it is also valid to set $t_F = 0$ to obtain the optimization problem.

(b) This problem is an LP since it has a linear objective and linear constraints. Define the vector of optimization variables as, $x = \begin{bmatrix} t_F \\ t_C \\ t_K \\ t_{Pl} \\ t_{Pa} \\ t_V \end{bmatrix}$.

Then we express this in standard form by defining $c = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ and $d = 2$ so the objective becomes,

$$\min_x c^\top x + d \tag{5}$$

The constraints become,

$$\underbrace{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} t_F \\ t_C \\ t_K \\ t_{Pl} \\ t_{Pa} \\ t_V \end{bmatrix}}_{x} \leq \underbrace{\begin{bmatrix} -3 \\ -3 \\ -3 \\ -1 \\ -2 \\ -3 \\ -3 \\ 0 \end{bmatrix}}_{b} \tag{6}$$

In this question we have asked you to clearly identify the relevant vectors/matrices just to be explicit; however this level of detail is not required for all such questions if you write clearly write the program in such a way it fits into one of the standard classes (unless we explicitly ask for it).

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

$$\min_{L,R} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k}, \;\; R \in \mathbb{R}^{p,k}. \tag{7}$$

(a) What is the largest possible rank of the matrix $LR^\top$ referenced in Equation (7)?

(b) Find a solution to the problem in Equation (7). 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

$$\min_{L,R} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k}, \;\; R \in \mathbb{R}^{p,k}, \;\; L \geq 0, \;\; R \geq 0 \tag{8}$$

You might wonder if the problem in Equation (8) 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 (8) is quite difficult to solve in general. In many cases, we will consider *alternating minimization* algorithms on the objective in Equation (8). 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

$$\min_{L} \|X - LR^\top\|_F^2 \quad : \quad L \in \mathbb{R}^{n,k} \;\; R \in \mathbb{R}^{p,k}, \;\; L \geq 0, \tag{9}$$

(and vice-versa to optimize over R). Can the problem in (9) 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 (9) with respect to $L \in \mathbb{R}^n$.

Solution 2

(a) The largest possible rank of $LR^\top$ is k. For more details see the Quiz 1 Problem 2 solutions.

(b) Note $LR^\top$ has maximum possible rank k but there are no other constraints on R, L. However the best rank k "approximation" to a matrix X in the Frobenius norm is simply given by truncating its singular value decomposition at the kth singular value. Formally if we take the SVD, $X = USV^\top$ (where the singular values of S on its diagonal are sorted in decreasing order) then the matrix $U\tilde{S}_kV^\top$, where $\tilde{S}_k$ is given by zeroing all but the top k singular values in S, solves the aforementioned optimization problem.

We can construct a suitable pair of L, R by simply taking $L = U\tilde{S}_k$ and $R = V$.

(c) If we consider the simple case presented where X, L, R are all scalars we can see that for any positive constant $c > 0$,

$$L = c\sqrt{X}, R = \frac{1}{c}\sqrt{X} \tag{10}$$

[1] This gradient can be useful to derive yet another iterative optimization algorithm to solve Equation (9) itself.

is a solution – so the solution is not unique. The above construction also generalizes to the matrix case easily although this was not required for the problem.

(d) The problem is in fact a convex QP. Let the rows of the matrix X be indexed by x_i and the rows of the matrix L be indexed by vectors l_i.

Then, note the Frobenius decouples over the rows of the matrix L in the sense that,

$$\min_L \|X - LR^\top\|_F^2 = \min_{l_i} \sum_{i=1}^{n} \|x_i - Rl_i\|_2^2 = \sum_{i=1}^{n} \min_{l_i} \|x_i - Rl_i\|_2^2 = \tag{11}$$

$$\sum_{i=1}^{n} \min_{l_i} \{(l_i^\top R^\top R l_i) - 2x_i^\top R l_i + \|x_i\|_2^2\} \tag{12}$$

so the objective decouples into several independent quadratic objectives (in l_i). Note that in the above we have taken the convention that x_i is $p \times 1$ and l_i is $k \times 1$ to express the problem in our usual conventions (the above terms are indeed all scalars).

The constraints $L \geq 0$ are simply that $l_i \geq 0$ for all $i \in \{1, \ldots, n\}$ which are linear constraints.

So the problem can either be formulated as the solution to several independent QP's or cast as one large QP by concatenating the l_i into one optimization variable x and defining the standard form vectors/matrices appropriately (although this final explicit description was not necessary for the problem as long as a clear description along the lines of the above was provided).

(e) For the following we assume that $L, R^\top$ are the vectors as given in the problem statement (they are $n \times 1$ and $1 \times p$ respectively). Recalling that $\|A\|_F^2 = \operatorname{trace}(A^\top A)$ and using the cyclic properties of the trace, we can expand the objective as,

$$f(l) = \|X - LR^\top\|_F^2 = \|X\|_F^2 - 2\operatorname{trace}(RL^\top X) + \operatorname{trace}(RL^\top LR^\top) = \tag{13}$$

$$\|X\|_F^2 - 2R^\top X^\top L + \|L\|_2^2 \|R\|_2^2 \tag{14}$$

Differentiating the above expression with respect to L gives the gradient as,

$$\nabla_L f(L) = -2XR + 2\|R\|_2^2 L \tag{15}$$