EE 127 2/20/14
L. El Ghaoui

Homework Assignment #3

Due date: 3/6/14, in class.

Exercise 1 (Squaring SOCP constraints) When considering a second-order cone constraint, a temptation might be to square it in order to obtain a classical convex quadratic constraint. This might not always work. Consider the constraint

$$x_1 + 2x_2 \geq \|x\|_2,$$

and its squared counterpart:

$$(x_1 + 2x_2)^2 \geq \|x\|_2^2.$$

Is the set defined by the second inequality convex? Discuss.

Exercise 2 (A complicated function) We would like to minimize the function $f : \mathbb{R}^3 \to \mathbb{R}$, with values:

$$\begin{aligned} f(x) &= \max\big(x_1 + x_2 - \min\big(\min(x_1 + 2, x_2 + 2x_1 - 5), x_3 - 6), \\ &\qquad \frac{(x_1 - x_3)^2 + 2x_2^2}{1 - x_1}\big), \end{aligned}$$

with the constraint $\|x\|_\infty < 1$. Explain precisely how to formulate the problem as an SOCP in standard form.

Exercise 3 (A minimum time path problem) Consider Figure 1, in which a point in 0 must move to reach point $p = [4 \quad 2.5]^\top$, crossing three layers of fluids having different densities.

In the first layer, the point can travel at a maximum speed v_1, while in the second layer and third layers it may travel at a lower maximum speeds, respectively $v_2 = v_1/\eta_2$, and $v_3 = v_1/\eta_3$, with $\eta_2, \eta_3 > 1$. Assume $v_1 = 1$, $\eta_2 = 1.5$, $\eta_3 = 1.2$. You have to determine what is the fastest (i.e., minimum time) path from 0 to p. *Hint:* you may use path leg lengths ℓ_1, ℓ_2, ℓ_3 as variables, and observe that, in this problem, equality constraints of the type $\ell_i =$ "something" can be equivalently substituted by inequality constraints $\ell_i \geq$ "something" (explain why).

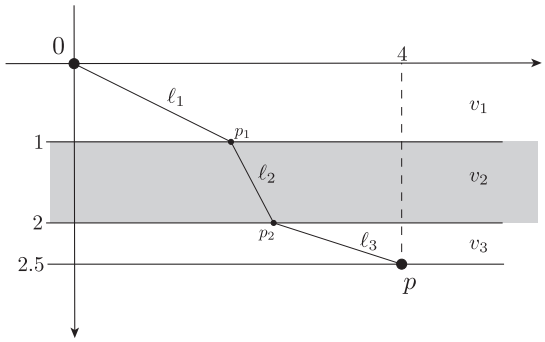

Figure 1: A minimum-time path problem.

Exercise 4 (A portfolio design problem) The returns on $n = 4$ assets are described by a Gaussian (Normal) random vector $r \in \mathbb{R}^n$, having the following expected value $\hat{r}$ and covariance matrix Σ:

$$\hat{r} = \begin{bmatrix} 0.12 \\ 0.10 \\ 0.07 \\ 0.03 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.0064 & 0.0008 & -0.0011 & 0 \\ 0.0008 & 0.0025 & 0 & 0 \\ -0.0011 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The last (fourth) asset corresponds to a risk-free investment. An investor wants to design a portfolio mix with weights $x \in \mathbb{R}^n$ (each weight x_i is nonnegative, and the sum of the weights is one) so to obtain the best possible expected return $\hat{r}^\top x$, under a set of conditions. Consider the following constraints:

(i) no single asset weights more than 40%;

(ii) the risk-free assets should not weight more than 20%;

(iii) no asset should weight less than 5%;

(iv) The Sharpe ratio (ratio of expected return to standard deviation of return) should be above 1.5.

(v) the probability of experiencing a return lower than $q = -3\%$ should be no larger than $\epsilon = 10^{-4}$.

What is the maximal achievable expected return under constraints (i)-(iv) OR (i)-(iii) and (v)? (*You only need to pick one set of constraints for full-credit. If you implement all 5 constraints successfully, you will receive an additional point of extra credit.*)

Exercise 5 (Sparse Classification, Word Imaging, and Support Vector Machines) The image of a given query word in a given corpus of text news can be defined as a short list of other words with which this query is strongly associated. To be easily understandable, the list should be extremely short with respect to the dictionary of terms present in the corpus. One way to obtain a word image is to use L1 penalized classification algorithm, where indicator of the query words appearance in each headline is used as that headlines label/response ($y \in \{1, -1\}^m$), and the indicators for all other words are used as predictors/features ($X = [x_1, \ldots, x_m] \in R^{n \times m}$). A standard classification algorithm seeks to linearly separate the data points with different labels via a hyper plane $H(w, b) := \{x : w^T x + b = 0\}$, where $w \in R^n/\{0\}$ and $b \in R$ are the parameters of the classifier. Precisely, we wish to find, if possible, (w, b) so that $w^T x + b > 0$ when $y_i = +1$ and $w^T x + b < 0$ otherwise. To evaluate the performance of a given classifier (w, b), we use a loss function that measures the number of errors on the training set, that is:

$$L(w, b) := \sum_{i: y_i(w^T x_i + b) < 0}^{m} 1$$

and we want to find (w, b) that minimize the error. Since L is not convex function we can't solve the problem directly, however we can upper bound the loss function by the surrogate loss function:

$$\tilde{L}(w, b) := \sum_{i=1}^{m} [1 - y_i(w^T x_i + b)]_+$$

where $z_+ := \max(0, z)$

By imposing a sparsity constraint on the weight vector, we can single out the few words that are most able to predict the presence or absence of a query word in any document. These selected words are then considered the list of words comprising the query words image. So we consider surrogate loss function with L1 regularization:

$$\min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \sum_{i=1}^{m} \left[1 - y_i(w^T x_i + b)\right]_+ + \lambda \|w\|_1$$

The above problem is also called L1- Support Vector Machine (L1-SVM).

1. Show how to formulate L1-SVM as a linear program.

2. We look at the Word Imaging problem in a small-scale setting. Our original data is the headlines of New York Times between Jan 1 2006 and Dec 31 2006: there are 84612 headlines and 160,624 distinct words in total. To make the problem accessible to you, we preprocessed the text data and down-sampled (with special care) both the number of distinct words and headlines to 997 and 1045, respectively. We try to obtain the image of the query word "Microsoft". The label y, the predictor X and the dictionary of terms dict are defined in **sparseSVM.mat**. Note, y_j indicates whether "Microsoft"

shows up in j^{th} headline, X_{ij} indicates whether j^{th} headline contains i^{th} word, and $dict_i$ is the i^{th} word. Using CVX, write a MATLAB program to solve the problem of minimizing the approximate loss for the given data. What are the top 20 words (i.e. top 20 features with highest coefficients) that predict the presence of the query word "Microsoft"? Does that make sense to you? Experiment with $\lambda \in \{0.1, 0.5, 1, 5, 10\}$ and see how the list of top words changes.

3. Consider now the traditional Support Vector Machine setup where L2 regularization is used (call it the L2-SVM)

$$\min_{\substack{w\in\mathbb{R}^n \\ b\in\mathbb{R}}} \sum_{i=1}^{m} \left[1 - y_i(w^T x_i + b)\right]_+ + \lambda \|w\|_2$$

   Show how to formulate this as an SOCP.

4. Follow the procedure from part 2 to solve this problem using the same lambdas. Discuss your results.

5. Compare the top 20 features extracted by the L1-SVM and the L2-SVM for each lambda. How do they compare across the lambdas? Which formulation makes more sense to you and why?