Midterm
Due: in class on 3/21/2013

Exam rules and guidelines

- You must turn in the exam at the **beginning** of class on 21st March, 2013. If you fail to turn in the exam at that point, there will be a loss of attainable points at a geometric rate (i.e., for every delayed hour, your possible max score will shrink by a factor of 0.95.)
- You may use any books, notes, software (Matlab, CVX, CVXOPT, Python, Sage, Mathematica, Maple, etc.), but you **may not discuss** the exam with anybody else. The only exception is that you are allowed to email us (the GSIs or the instructor) in case you need clarifications.
- None of the solutions is supposed to be long and tedious—in case you find that your solution is becoming rather long, please double think, and try to simplify.
- **Start each problem on a new page and put together your solutions in sequential order**
- Preferably LaTeXyour solutions. If you cannot LaTeXthem, make **sure that your handwriting is super legible; if we cannot read it easily, that'll lead to a loss of points.** We expect the same quality of work even from those enrolled on Pass/Fail basis.
- Please make a copy of your exam before you hand it in (especially, in case you are turning in a handwritten exam).
- In case any of your answers requires running computation, please attach source-code and final numerical results of running your code.

Total credit: 70pts + 5 bonus pts (in case you need 'em).

1. **Convex sets and functions: [15pts]**

   (a) If $f : \mathbb{R}_+ \to \mathbb{R}_+$ is concave and $f(0) = 0$, then f must be subadditive (i.e., $f(x+y) \leq f(x)+f(y)$)

   Proof. We have

$$f(\alpha) = f(\frac{\alpha}{\alpha+\beta}(\alpha+\beta) + \frac{\beta}{\alpha+\beta}0) \geq \frac{\alpha}{\alpha+\beta}f(\alpha+\beta) + \frac{\beta}{\alpha+\beta}f(0)$$
$$f(\beta) = f(\frac{\beta}{\alpha+\beta}(\alpha+\beta) + \frac{\alpha}{\alpha+\beta}0) \geq \frac{\beta}{\alpha+\beta}f(\alpha+\beta) + \frac{\alpha}{\alpha+\beta}f(0),$$

   from which upon using $f(0) = 0$ it immediately follows that

$$f(\alpha) + f(\beta) \geq f(\alpha+\beta).$$

   □

(b) Prove that the function $f(x) = -\log(\cos x)$ is convex on $(-\pi/2, \pi/2)$. Find its conjugate f^*. What is dom f^*?

We see that $f''(x) = \sec^2(x) = \frac{2}{\cos(2x)+1}$. Notice that $\cos(x) > 0$ on $(-\pi/2, \pi/2)$, hence the effective domain of f may be treated as $(-\pi/2, \pi/2)$ (of course, since cos is a periodic function, f is actually convex in other regions too, but for simplicity $(-\pi/2, \pi/2)$ suffices).

The conjugate of f is given by $f^*(y) = y\tan^{-1}(y) - \frac{1}{2}\log(1+y^2)$. Clearly, this function has domain $\mathbb{R}$.

(c) Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a symmetric function, (i.e., if $x = [x_1, x_2, \ldots, x_n]$ and $x_\sigma = [x_{\sigma(1)}, \ldots, x_{\sigma(n)}]$ for any permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, then $f(x_\sigma) = f(x)$). Let $S^{n\times n}$ be the set of $n \times n$ symmetric matrices, and $\lambda : S^{n\times n} \to \mathbb{R}^n$ the *eigenvalue map*, that maps a symmetric matrix to the sorted ($\downarrow$) vector of its eigenvalues. Show that

$$(f \circ \lambda)^* = f^* \circ \lambda.$$

[*Hint:* This question is simpler than it appears. Use the fact that for any two matrices $X, Y \in S^{n\times n}$ we have the inequality

$$\mathrm{Tr}(XY) \le \lambda(X)^T\lambda(Y).$$

Also useful is to remember that $\lambda(\cdot)$ and Tr enjoy the following invariance: $\lambda(QAQ^T) = \lambda(A)$ for orthogonal Q, and $\mathrm{Tr}(QAQ^T) = \mathrm{Tr}(A)$. To prove the claim, try showing $(f \circ \lambda)^* \le f^* \circ \lambda$ and $(f \circ \lambda)^* \ge f^* \circ \lambda$. It'll be helpful to consider $Y = U\mathrm{Diag}\lambda(Y)U^T$.]

Using the hint, we first show that $(f \circ \lambda)^* \le f^* \circ \lambda$. To that end,

$$\begin{aligned}
[f(\lambda(Z))]^* &= \sup_X \quad \mathrm{Tr}(XZ) - f(\lambda(X)) \\
&\overset{\text{hint}}{\le} \sup_X \quad \lambda(X)^T\lambda(Z) - f(\lambda(X)) \\
&= f^*(\lambda(Z)).
\end{aligned}$$

To prove the other direction, first write the eigenvector decomposition: $Z = U\,\mathrm{Diag}(\lambda(Z))U^T$ (so that $U^TZU = \mathrm{Diag}(\lambda(Z))$). Then, (notice the step with the inequality uses symmetry of f):

$$\begin{aligned}
f^*(\lambda(Z)) &= \sup_{x\in\mathbb{R}^n} \quad \big(x^T\lambda(Z) - f(x)\big) \\
&= \sup_{x\in\mathbb{R}^n} \big(\mathrm{Tr}(\mathrm{Diag}(x)U^TZU) - f(x)\big) \\
&= \sup_{x\in\mathbb{R}^n} \big(\mathrm{Tr}(U\,\mathrm{Diag}(x)U^TZ) - f(\lambda(U\,\mathrm{Diag}(x)U^T))\big) \\
&\le \sup_{X\in S^{n\times n}} \big(\mathrm{Tr}(XZ) - f(\lambda(X))\big) \\
&= (f \circ \lambda)^*(Z).
\end{aligned}$$

(d) *Bonus****:* Let $f : I \to (0, \infty)$, where I is an interval. Prove that $\log f$ is convex if and only if, $e^{cx}f(x)$ is convex (in x) on the interval I for all $c \in \mathbb{R}$.

If $\log f$ is convex, then $cx + \log f(x)$ is convex for every c (linear functions are convex). The function $x \mapsto e^x$ is increasing and convex, it follows from the composition rules (Ch. 3 of textbook) that $e^{cx+\log f(x)} = e^{cx}f(x)$ is convex.

The converse direction is slightly trickier. Assume that $e^{cx}f(x)$ is convex for every $c \in \mathbb{R}$. For every closed subinterval $J \subset I$, the maximum of $e^{cx}f(x)$ for an $x \in J$ is for an x on some endpoint of J. This means that the maximum of $\log(e^{cx}f(x)) = cx + \log f(x)$ is also taken when x is some endpoint. Thus, $cx + \log f(x)$ within J lies below the values at the endpoints. Since J was arbitrary, this implies convexity of $cx + \log(x)$.

2. **Subdifferentials: [10pts]** Find a single subgradient of the following functions:

   (a) $f(x) = \sum_{i=1}^{k} |x|_{[i]}$, where $|x|_{[i]}$ denotes the i-th largest absolute value in x. (Here, $k \leq n$ is given.)
   Solution: We first note that the sum of k-largest absolute values can be written in the following variational form:

$$\sum_{i=1}^{k} |x|_{[i]} = \max_{z \in \{-1,0,1\}^n \ card(z) \leq k} x^T z$$

   Therefore applying the subgradient rule for maximum, we get:

$$\partial f = Conv\{z \ : \ z_{[i]} = sign(x_{[i]}), \ for \ i \in 1, ..., k \ and \ z_j = 0 \ o.w.\}$$

   Therefore a single subgradient is given by setting $z_j = 0$ except $z_{[i]} = sign(x_{[i]}), \ for \ i \in 1, ..., k$.

   (b) $f(x) = \lambda_{max}(e^{-\sum_{i=1}^{n} A_i x_i})$, where λ_{max} is the maximum eigenvalue and A_i are fixed $n \times n$ matrices.
   Hint: e^X here refers to the matrix exponential: $e^X = \sum_{k \geq 0} \frac{X^k}{k!}$, and from this definition it follows that $\frac{d}{dt} e^{tA} = Ae^{tA}$
   Solution: We first let $A(x) = \sum_{i=1}^{n} A_i x_i$ and then write the maximum eigenvalue in variational form as follows

$$f(x) = \max_{||y||_2 \leq 1} y^T e^{-A(x)} y$$

   Therefore applying the subgradient rule for maximum, we get:

$$\partial f = Conv\{(-y^T A_1 e^{-A(x)} y, ..., -y^T A_n e^{-A(x)} y) \ : \ y = \arg \max_{||z||_2 \leq 1} z^T e^{-A(x)} z\}$$

$$\partial f = Conv\{(-y^T A_1 e^{-A(x)} y, ..., -y^T A_n e^{-A(x)} y) \ : \ y \text{ is an eigenvector for one of the maximum eigenvalues}\}$$

   Therefore a single subgradient is given by $(-y^T A_1 e^{-A(x)} y, ..., -y^T A_n e^{-A(x)} y)$ where y is a maximal eigenvector.

3. **Optimization problems: [10pts]**

   (a) [2pts] Prove that

$$\inf_{x,y} F(x, y) = \inf_{x} \left(\inf_{y} F(x, y) \right).$$

   [Obvious!]

   (b) [4pts] Suppose $G \in \mathbb{R}^{n \times n}$ is an input matrix. Write down the following optimization problem as an SDP (explain the steps)

$$\min \quad \tfrac{1}{2} \|X - G\|_{\mathrm{F}}^2 \qquad X \succeq 0, \quad X_{ii} = 1.$$

   (Note: $\|Z\|_{\mathrm{F}}^2 = \mathrm{Tr}(Z^T Z)$ for an arbitrary matrix).
   Solution: The key observation is that $\|Z\|_{\mathrm{F}} = \|\mathrm{vec}(Z)\|_2$, where the vec operator just stacks columns of Z into a long vector. Notice that vec is just a linear operator. So we may write the above problem as

$$\min_{X,t} \quad t$$
$$\|\mathrm{vec}(X - G)\|_2 \leq t, \quad X \succeq 0, X_{ii} = 1, \ 1 \leq i \leq n.$$

   The first constraint is an SOCP constraint (since vec is a linear operator), which we know how to write as a semidefinite constraint, namely as

$$\begin{bmatrix} t & \mathrm{vec}(X - G)^T \\ \mathrm{vec}(X - G) & tI \end{bmatrix} \succeq 0.$$

(c) [4pts] The operator ℓ_1-norm of a matrix $A \in \mathbb{R}^{m\times n}$ is defined as

$$\|A\|_1 := \sup_{x\neq 0} \frac{\|Ax\|_1}{\|x\|_1} = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|,$$

that is, the maximum absolute column sum. Suppose, you have data matrices $B_1, \ldots, B_k \in \mathbb{R}^{m\times n}$, and you wish to find a vector $x \in \mathbb{R}^k$ that helps best approximate a target matrix A by minimizing $\left\|A - \sum_{i=1}^{k} x_i B_i\right\|_1$ (over $x \in \mathbb{R}^k$). Formulate the above task as a linear program, and explain how you can recover a solution to the original problem from a solution to your LP.

Solution: We can write,

$$p^* = \min_{x,Z} \max_j \sum_i |Z_{ij}| \;:\; Z = A - \sum_i x_i B_i = \min_{x,y,Z} y \;:\; y \geq \sum_i |Z_{ij}| \;\; \forall j, \;\; Z = A - \sum_i x_i B_i, \tag{1}$$

$$= \min_{x,y,Z,W} y \;:\; y \geq \sum_i W_{ij} \;\; \forall j, \;\; Z = A - \sum_i x_i B_i, \;\; Z_{ij} \leq W_{ij}, \;\; -Z_{ij} \leq W_{ij} \tag{2}$$

which is an LP in variables x,y,Z,W. We can recover the solution to the original problem using the variable x, and forming the approximation in ℓ_1 operator norm by $A \approx \sum_i x_i B_i$.

4. **Duality and Optimality: [10pts]**

(a) Find the dual of the following problem,

$$\min_x \|Ax - b\|_2 + \lambda \|x\|_1$$

Solution: We use the dual representations of 2-norm and 1-norm as follows.

$$p^* = \min_x \max_{||y||_2 \leq 1, \; ||v||_\infty \leq \lambda} y^T(Ax - b) + v^T x$$

The dual is given by interchanging min and max and eliminating x:

$$d^* = \max_{||y||_2 \leq 1, \; ||v||_\infty \leq \lambda} -y^T b \;:\; A^T y + v = 0 \tag{3}$$

$$= \max_{||y||_2 \leq 1, \; ||A^T y||_\infty \leq \lambda} -y^T b \tag{4}$$

Also note that strong duality holds since the problem is convex and dual is strictly feasible.

(b) Using optimality conditions of the above problem, show that if the i'th column of matrix A satisfies $\|a_i\|_2 < 1$ then $x_i^* = 0$ at the optimum.

Solution: The dual feasibility condition requires that $||A^T y||_\infty \leq \lambda$. If for some i we have $|a_i^T y^*| < \lambda$ at optimum then by complementary slackness $x_i^* = 0$. Another way to think is that, if the $i'th$ constraint, $|a_i^T y^*| \leq \lambda$, is not active at optimum we can remove that constraint (effectively removing $i'th$ column of A) without changing the optimal value, which implies that $x_i^* = 0$. Now note that $||y^*||_2 \leq 1$ and Cauchy-Schwartz imply that,

$$|a_i^T y^*| \leq ||a_i||_2 ||y^*||_2 \leq ||a_i||_2$$

which proves that $||a_i||_2 < 1$ implies $x_i^* = 0$. In high-dimensional problems, this inequality offers a very efficient way to pre-process the data and eliminate the variables in the problem which are guaranteed to be 0 at optimum.

5. **Algorithms [25pts]** Consider the optimization problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \left(\tfrac{1}{2} d_i x_i^2 + r_i x_i\right) \\ \text{subject to} \quad & a^T x = 1, x_i \in [-1, 1] \quad i = 1, \ldots, n, \end{aligned}$$

where $a \neq 0$ and each $d_i > 0$.

(a) [4pts] Write down the KKT optimality conditions for this problem
Solution: By introducing necessary dual variables, we first write Lagrangian:

$$\sum_i \frac{1}{2} d_i x_i^2 + r_i x_i + \theta(a^T x - 1) + \sum_i \lambda_i(-x_i - 1) + \sum_i \mu_i(x_i - 1)$$

Then the KKT conditions are given by,

i. Lagrangian stationarity: $d_i x_i + r_i + \theta a_i - \lambda_i + \mu_i = 0$ for $1 \le i \le n$
ii. Complementary slackness: $\lambda_i(-x_i - 1) = 0, \qquad \mu_i(x_i - 1) = 0$
iii. Primal feasibility: $a^T x = 1, \ -1 \le x_i \le 1$
iv. Dual feasibility: $\lambda_i \ge 0, \mu_i \ge 0$ for $1 \le i \le n$

(b) [1pts] Does strong duality hold in this problem?
Solution: Strong duality may or may not hold, depending on the constraint $a^T x = 1$. By Slater's condition, if this set is strictly feasible, then strong duality holds.

(c) [15pts]Use the KKT conditions and / or the Lagrangian to come up with the fastest algorithm you can to solve this optimization problem.
One simple way to solve this problem is to start with the Lagrangian. Then, introducing a new variable $y = D^{1/2}x$, we may rewrite the problem as

$$\min \quad \tfrac{1}{2} y^T y + \bar{r}^T y, \quad \bar{l} \le y \le \bar{u}, \quad \bar{a}^T y = 1,$$

where $\bar{r}_i = y_i/\sqrt{d_i}$, $\bar{l}_i = \sqrt{d_i} l_i$, $\bar{u}_i = \sqrt{d_i} u_i$ (in our case, $l_i = -1, u_i = 1$), and $\bar{a}_i = a_i/\sqrt{d_i}$. Now optimize the Lagrangian

$$L(y, \theta) := \inf \tfrac{1}{2} y^T y + \bar{r}^T y + \theta(\bar{a}^T y - 1),$$

subject to the constraint $\bar{l} \le y\bar{u}$. For a fixed value of θ (which we is yet unknown), we complete the square on the Lagrangian, and see that it is simply a projection problem onto the box $\bar{l} \le y \le \bar{u}$. Doing this projection and using $y = D^{1/2}x$, we ultimately obtain

$$x(\theta) = \min\left\{\max\left\{l, D^{-1}(a + \theta r)\right\}, u\right\}.$$

We need to obtain an optimal θ; to that end recall that we have the constraint $a^T x(\theta) = 1$ that must be satisfied. Using the KKT conditions, we can show that $x^* = x(\theta)$ iff $a^T x(\theta) = 1$. Thus, all we need to now do is to somehow compute the solution to the equation

$$g(\theta) := a^T x(\theta) - 1 = 0.$$

We can solve the above equation using MATLAB's `fzero` function. More directly, we can search for the optimal θ by studying $g(\theta)$ more closely—e.g., plotting $g(\theta)$ we see that it has breakpoints given by $(a_i \pm d_i)/r_i$. Each $x_i(\theta)$ can be expressed in terms of the breakpoints as

$$x_i(\theta) = \begin{cases} 1 & \text{if } \ \theta \le (a_i - d_i)/r_i \\ (a_i + \theta r_i)/d_i & \text{if } \ (a_i - d_i)/r_i \le \theta \le (a_i + d_i)/r_i \\ -1 & \text{if } \ \theta \ge (a_i + d_i)/r_i. \end{cases}$$

Thus, $g(\theta)$ is piecewise continuous and nonincreasing. We just need to find an interval bracketing θ^*, which can be done by using a median-search over the intervals containing these breakpoints. This requires slightly careful programming. Alternatively, we can extract upper and lower bounds on θ from the above breakpoints and simply do binary-search.

(d) [5pts] Analyze the running time complexity of your algorithm. Does the empirical performance of your method agree with your analysis? Assuming we do the binary search as mentioned above. Then, each evaluation of $x(\theta)$ takes linear time $O(n)$ to compute. Thus, the binary search idea runs in pseudo-linear time, taking precisely $O(n \log(\frac{\theta_{\max} - \theta_{\min}}{\epsilon}))$ iterations to obtain the root to tolerance ϵ. Of course, if we use the cleverer algorithm hinted at, the complexity can be made independent of ϵ, and is in fact $O(n)$.