20 Lecture 20, Mar 10

Announcements

- HW6 (EM/MM, handwritten digit recognition revisited) due Fri Mar 11 @ 11:59PM.
- Solution sketches for HW1-5 are posted. `http://hua-zhou.github.io/teaching/biostatm280-2016winter/hwXXsol.html`. Substitute XX by 01, 02, ...
- Quiz 4 today.
- Course evaluation: `http://my.ucla.edu`.

Last time

- Linear programming (LP): more examples.
- Quadratic programming (QP).

Today

- Second order cone programming (SOCP).
- Semidefinite programming (SDP).
- Geometric programming (GP).
- Conclusion remarks.

Second-order cone programming (SOCP)

- A *second-order cone program* (SOCP)

$$\begin{array}{ll} \text{minimize} & \boldsymbol{f}^T \boldsymbol{x} \\ \text{subject to} & \|\boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{b}_i\|_2 \le \boldsymbol{c}_i^T \boldsymbol{x} + d_i, \quad i = 1, \ldots, m \\ & \boldsymbol{F}\boldsymbol{x} = \boldsymbol{g} \end{array}$$

over $\boldsymbol{x} \in \mathbf{R}^n$. This says the points $(\boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i, \boldsymbol{c}_i^T\boldsymbol{x} + d_i)$ live in the second order cone (ice cream cone, Lorentz cone, quadratic cone)

$$\mathbf{Q}^{n+1} = \{(\boldsymbol{x}, t) : \|\boldsymbol{x}\|_2 \le t\}$$

in $\mathbf{R}^{n+1}$.

☞ QP is a special case of SOCP. Why?

- When $\boldsymbol{c}_i = \mathbf{0}$ for $i = 1, \ldots, m$, SOCP is equivalent to a *quadratically constrained quadratic program* (QCQP)

$$\begin{array}{ll} \text{minimize} & (1/2)\boldsymbol{x}^T\boldsymbol{P}_0\boldsymbol{x} + \boldsymbol{q}_0^T\boldsymbol{x} \\ \text{subject to} & (1/2)\boldsymbol{x}^T\boldsymbol{P}_i\boldsymbol{x} + \boldsymbol{q}_i^T\boldsymbol{x} + r_i \le 0, \quad i = 1, \ldots, m \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \end{array}$$

  where $\boldsymbol{P}_i \in \mathbf{S}_+^n$, $i = 0, 1, \ldots, m$. Why?

- A *rotated quadratic cone* in $\mathbf{R}^{n+2}$ is

$$\mathbf{Q}_r^{n+2} = \{(\boldsymbol{x}, t_1, t_2) : \|\boldsymbol{x}\|_2^2 \le 2t_1t_2, t_1 \ge 0, t_2 \ge 0\}.$$

  A point $\boldsymbol{x} \in \mathbf{R}^{n+1}$ belongs to the second order cone $\mathbf{Q}^{n+1}$ if and only if

$$\begin{pmatrix} \boldsymbol{I}_{n-2} & 0 & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \boldsymbol{x}$$

  belongs to the rotated quadratic cone $\mathbf{Q}_r^{n+1}$.

  ☞ Gurobi allows users to input second order cone constraint and quadratic constraints directly.

  ☞ Mosek allows users to input second order cone constraint, quadratic constraints, and rotated quadratic cone constraint directly.

- Following sets are *(rotated) quadratic cone representable sets*:
  - (Absolute values) $|x| \le t \Leftrightarrow (x, t) \in \mathbf{Q}^2$.
  - (Euclidean norms) $\|\boldsymbol{x}\|_2 \le t \Leftrightarrow (\boldsymbol{x}, t) \in \mathbf{Q}^{n+1}$.

– (Sume of squares) $\|\boldsymbol{x}\|_2^2 \le t \Leftrightarrow (\boldsymbol{x}, t, 1/2) \in \mathbf{Q}_r^{n+2}$.

– (Ellipsoid) For $\boldsymbol{P} \in \mathbf{S}_+^n$ and if $\boldsymbol{P} = \boldsymbol{F}^T\boldsymbol{F}$, where $\boldsymbol{F} \in \mathbf{R}^{n\times k}$, then

$$\begin{aligned} & (1/2)\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} + r \le 0 \\ \Leftrightarrow\quad & \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} \le 2t, t + \boldsymbol{c}^T\boldsymbol{x} + r = 0 \\ \Leftrightarrow\quad & (\boldsymbol{F}\boldsymbol{x}, t, 1) \in \mathbf{Q}_r^{k+2}, t + \boldsymbol{c}^T\boldsymbol{x} + r = 0. \end{aligned}$$

Similarly,

$$\|\boldsymbol{F}(\boldsymbol{x} - \boldsymbol{c})\|_2 \le t \Leftrightarrow (\boldsymbol{y}, t) \in \mathbf{Q}^{n+1}, \boldsymbol{y} = \boldsymbol{F}(\boldsymbol{x} - \boldsymbol{c}).$$

☞ This fact shows that QP and QCQP are instances of SOCP.

– (Second order cones) $\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\|_2 \le \boldsymbol{c}^T\boldsymbol{x} + d \Leftrightarrow (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{c}^T\boldsymbol{x} + d) \in \mathbf{Q}^{m+1}$.

– (Simple polynomial sets)

$$\begin{aligned} \{(t,x) : |t| \le \sqrt{x}, x \ge 0\} &= \{(t,x) : (t, x, 1/2) \in \mathbf{Q}_r^3\} \\ \{(t,x) : t \ge x^{-1}, x \ge 0\} &= \{(t,x) : (\sqrt{2}, x, t) \in \mathbf{Q}_r^3\} \\ \{(t,x) : t \ge x^{3/2}, x \ge 0\} &= \{(t,x) : (x, s, t), (s, x, 1/8) \in \mathbf{Q}_r^3\} \\ \{(t,x) : t \ge x^{5/3}, x \ge 0\} &= \{(t,x) : (x, s, t), (s, 1/8, z), (z, s, x) \in \mathbf{Q}_r^3\} \\ \{(t,x) : t \ge x^{(2k-1)/k}, x \ge 0\}, k \ge 2, &\ \text{can} \ \text{be represented similarly} \\ \{(t,x) : t \ge x^{-2}, x \ge 0\} &= \{(t,x) : (s, t, 1/2), (\sqrt{2}, x, s) \in \mathbf{Q}_r^3\} \\ \{(t,x,y) : t \ge |x|^3/y^2, y \ge 0\} &= \{(t,x,y) : (x, z) \in \mathbf{Q}^2, (z, y/2, s), (s, t/2, z) \in \mathbf{Q}_r^3\} \end{aligned}$$

– (Geometric mean) The hypograph of the (concave) geometric mean function

$$\mathbf{K}_{\mathrm{gm}}^n = \{(\boldsymbol{x}, t) \in \mathbf{R}^{n+1} : (x_1x_2\cdots x_n)^{1/n} \ge t, \boldsymbol{x} \succeq \mathbf{0}\}$$

can be represented by rotated quadratic cones. See (Lobo et al., 1998) for derivation. For example,

$$\begin{aligned} \mathbf{K}_{\mathrm{gm}}^2 &= \{(x_1, x_2, t) : \sqrt{x_1x_2} \ge t, x_1, x_2 \ge 0\} \\ &= \{(x_1, x_2, t) : (\sqrt{2}t, x_1, x_2) \in \mathbf{Q}_r^3\}. \end{aligned}$$

– (Harmonic mean) The hypograph of the harmonic mean function $\left(n^{-1}\sum_{i=1}^n x_i^{-1}\right)^{-1}$ can be represented by rotated quadratic cones

$$\begin{aligned}
&\left(n^{-1}\sum_{i=1}^n x_i^{-1}\right)^{-1} \geq t, \boldsymbol{x} \succeq \mathbf{0} \\
\Leftrightarrow\quad & n^{-1}\sum_{i=1}^n x_i^{-1} \leq y, \boldsymbol{x} \succeq \mathbf{0} \\
\Leftrightarrow\quad & x_i z_i \geq 1, \sum_{i=1}^n z_i = ny, \boldsymbol{x} \succeq \mathbf{0} \\
\Leftrightarrow\quad & 2x_i z_i \geq 2, \sum_{i=1}^n z_i = ny, \boldsymbol{x} \succeq \mathbf{0}, \boldsymbol{z} \succeq \mathbf{0} \\
\Leftrightarrow\quad & (\sqrt{2}, x_i, z_i) \in \mathbf{Q}_r^3, \mathbf{1}^T\boldsymbol{z} = ny, \boldsymbol{x} \succeq \mathbf{0}, \boldsymbol{z} \succeq \mathbf{0}.
\end{aligned}$$

– (Convex increasing rational powers) For $p, q \in \mathbf{Z}_+$ and $p/q \geq 1$,

$$\mathbf{K}^{p/q} = \{(x,t) : x^{p/q} \leq t, x \geq 0\} = \{(x,t) : (t\mathbf{1}_q, \mathbf{1}_{p-q}, x) \in \mathbf{K}_{\text{gm}}^p\}.$$

– (Convex decreasing rational powers) For any $p, q \in \mathbf{Z}_+$,

$$\mathbf{K}^{-p/q} = \{(x,t) : x^{-p/q} \leq t, x \geq 0\} = \{(x,t) : (x\mathbf{1}_p, t\mathbf{1}_q, 1) \in \mathbf{K}_{\text{gm}}^{p+q}\}.$$

– (Power cones) The *power cone* with rational powers is

$$\mathbf{K}_{\boldsymbol{\alpha}}^{n+1} = \left\{(\boldsymbol{x}, y) \in \mathbf{R}_+^n \times \mathbf{R} : |y| \leq \prod_{j=1}^n x_j^{p_j/q_j}\right\},$$

where p_j, q_j are integers satisfying $0 < p_j \leq q_j$ and $\sum_{j=1}^n p_j/q_j = 1$. Let $\beta = \operatorname{lcm}(q_1, \ldots, q_n)$ and

$$s_j = \beta \sum_{k=1}^j \frac{p_k}{q_k}, \quad j = 1, \ldots, n-1.$$

Then it can be represented as

$$\begin{aligned}
&|y| \leq (z_1 z_2 \cdots z_\beta)^{1/q} \\
&z_1 = \cdots = z_{s_1} = x_1, \quad z_{s_1+1} = \cdots = z_{s_2} = x_2, \quad z_{s_{n-1}+1} = \cdots = z_\beta = x_n.
\end{aligned}$$

☞ References for above examples: Papers (Lobo et al., 1998; Alizadeh and Goldfarb, 2003) and book (Ben-Tal and Nemirovski, 2001, Lecture 3). Now our catalogue of SOCP terms includes all above terms.

☞ Most of these function are implemented as the built-in function in the convex optimization modeling language `cvx`.

- Example: Group lasso. In many applications, we need to perform variable selection at group level. For instance, in factorial analysis, we want to select or de-select the group of regression coefficients for a factor simultaneously. Yuan and Lin (2006) propose the group lasso that

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{y} - \beta_0 \mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2,$$

where $\boldsymbol{\beta}_g$ is the subvector of regression coefficients for group g, and w_g are fixed group weights. This is equivalent to the SOCP

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\boldsymbol{\beta}^T \boldsymbol{X}^T \left(\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) \boldsymbol{X}\boldsymbol{\beta} + \\ & \qquad \boldsymbol{y}^T \left(\boldsymbol{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) \boldsymbol{X}\boldsymbol{\beta} + \lambda \sum_{g=1}^{G} w_g t_g \\ \text{subject to} \quad & \|\boldsymbol{\beta}_g\|_2 \le t_g, \quad g = 1, \ldots, G, \end{aligned}$$

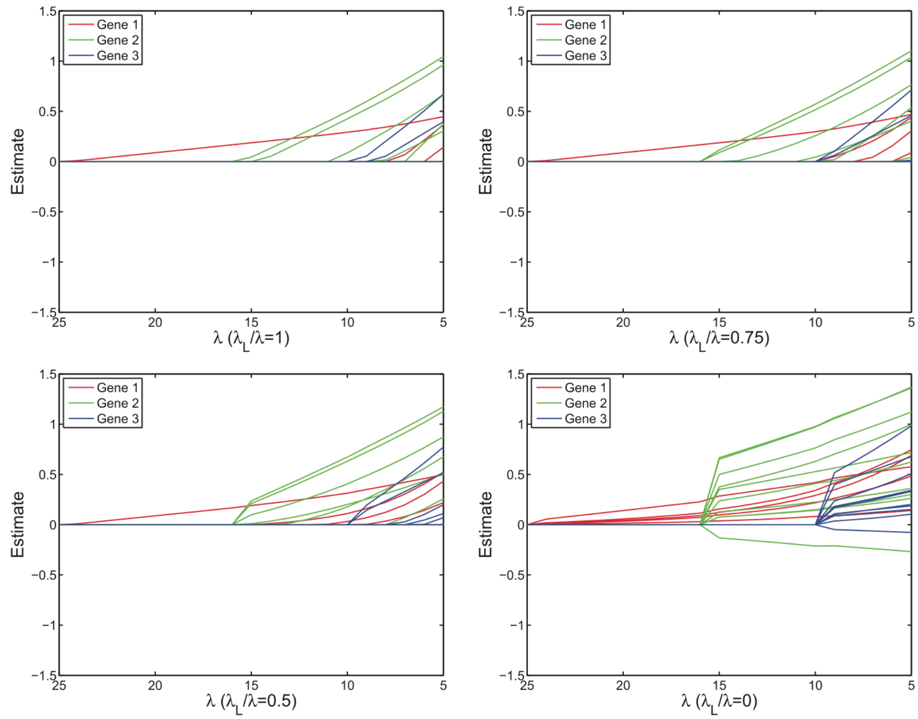
in variables $\boldsymbol{\beta}$ and $t_1, \ldots, t_G$.

☞ Overlapping groups are allowed here.

- Example. Sparse group lasso

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{y} - \beta_0 \mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{g=1}^{G} w_g \|\boldsymbol{\beta}_g\|_2$$

achieves sparsity at both group and individual coefficient level and can be solved by SOCP as well.



☞ Apparently we can solve any previous loss functions (quantile, ℓ_1, composite quantile, Huber, multi-response model) plus group or sparse group penalty by SOCP.

- Example. Square-root lasso (Belloni et al., 2011) minimizes

$$\|\boldsymbol{y} - \beta_0 \mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|_2 + \lambda \|\boldsymbol{\beta}\|_1$$

by SOCP. This variant generates the same solution path as lasso (why?) but simplifies the choice of λ.

A demo example: `http://hua-zhou.github.io/teaching/biostatm280-2016winter/lasso.html`

- Example: Image denoising by ROF model.

- Example. ℓ_p regression with $p \geq 1$ a rational number

$$\text{minimize} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_p$$

can be formulated as a SOCP. Why? For instance, $\ell_{3/2}$ regression combines advantage of both robust ℓ_1 regression and least squares.

☞ `norm(x, p)` is a built-in function in the convex optimization modeling language `cvx` and `Convex.jl`.

Semidefinite programming (SDP)

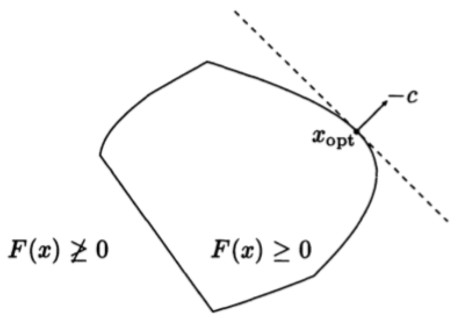


FIG. 1. *A simple semidefinite program with $x \in \mathbf{R}^2$, $F(x) \in \mathbf{R}^{7\times 7}$.*

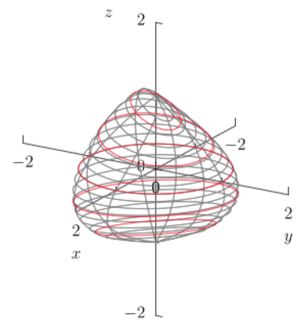


Figure 4.1: Plot of spectrahedron $S = \{(x, y, z) \in \mathbf{R}^3 \mid A(x, y, z) \succeq 0\}$.

- A *semidefinite program* (SDP) has the form

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^T \boldsymbol{x} \\ &\text{subject to} && x_1 \boldsymbol{F}_1 + \cdots + x_n \boldsymbol{F}_n + \boldsymbol{G} \preceq \mathbf{0} \quad \text{(LMI, linear matrix inequality)} \\ &&& \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \end{aligned}$$

  where $\boldsymbol{G}, \boldsymbol{F}_1, \ldots, \boldsymbol{F}_n \in \mathbf{S}^k$, $\boldsymbol{A} \in \mathbf{R}^{p\times n}$, and $\boldsymbol{b} \in \mathbf{R}^p$.

  ☞ When $\boldsymbol{G}, \boldsymbol{F}_1, \ldots, \boldsymbol{F}_n$ are all diagonal, SDP reduces to LP.

- The *standard form SDP* has form

$$\begin{aligned} &\text{minimize} && \operatorname{tr}(\boldsymbol{C}\boldsymbol{X}) \\ &\text{subject to} && \operatorname{tr}(\boldsymbol{A}_i \boldsymbol{X}) = b_i, \quad i = 1, \ldots, p \\ &&& \boldsymbol{X} \succeq \mathbf{0}, \end{aligned}$$

  where $\boldsymbol{C}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_p \in \mathbf{S}^n$.

- An *inequality form SDP* has form

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & x_1\boldsymbol{A}_1 + \cdots + x_n\boldsymbol{A}_n \preceq \boldsymbol{B}, \end{array}$$

  with variable $\boldsymbol{x} \in \mathbf{R}^n$, and parameters $\boldsymbol{B}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_n \in \mathbf{S}^n$, $\boldsymbol{c} \in \mathbf{R}^n$.

- Exercise. Write LP, QP, QCQP, and SOCP in form of SDP.

- Example. Nearest correlation matrix. Let $\mathbf{C}^n$ be the convex set of $n \times n$ correlation matrices

$$\mathbf{C} = \{\boldsymbol{X} \in \mathbf{S}_+^n : x_{ii} = 1, i = 1, \ldots, n\}.$$

  Given $\boldsymbol{A} \in \mathbf{S}^n$, often we need to find the closest correlation matrix to $\boldsymbol{A}$

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{A} - \boldsymbol{X}\|_{\mathrm{F}} \\ \text{subject to} & \boldsymbol{X} \in \mathbf{C}. \end{array}$$

  This projection problem can be solved via an SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\boldsymbol{A} - \boldsymbol{X}\|_{\mathrm{F}} \le t \\ & \boldsymbol{X} = \boldsymbol{X}^T, \, \mathrm{diag}(\boldsymbol{X}) = \mathbf{1} \\ & \boldsymbol{X} \succeq \mathbf{0} \end{array}$$

  in variables $\boldsymbol{X} \in \mathbf{R}^{n \times n}$ and $t \in \mathbf{R}$. The SOC constraint can be written as an LMI

$$\begin{pmatrix} t\boldsymbol{I} & \mathrm{vec}(\boldsymbol{A} - \boldsymbol{X}) \\ \mathrm{vec}(\boldsymbol{A} - \boldsymbol{X})^T & t \end{pmatrix} \succeq \mathbf{0}$$

  by the Schur complement lemma.

- Eigenvalue problems. Suppose

$$\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{A}_0 + x_1\boldsymbol{A}_1 + \cdots x_n\boldsymbol{A}_n,$$

  where $\boldsymbol{A}_i \in \mathbf{S}^m$, $i = 0, \ldots, n$. Let $\lambda_1(\boldsymbol{x}) \ge \lambda_2(\boldsymbol{x}) \ge \cdots \ge \lambda_m(\boldsymbol{x})$ be the ordered eigenvalues of $\boldsymbol{A}(\boldsymbol{x})$.

– Minimize the maximal eigenvalue is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \boldsymbol{A}(\boldsymbol{x}) \preceq t\boldsymbol{I} \end{array}$$

in variables $\boldsymbol{x} \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

☞ Minimizing the sum of k largest eigenvalues is an SDP too. How about minimizing the sum of all eigenvalues?

☞ Maximize the minimum eigenvalue is an SDP as well.

– Minimize the spread of the eigenvalues $\lambda_1(\boldsymbol{x}) - \lambda_m(\boldsymbol{x})$ is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & t_1 - t_m \\ \text{subject to} & t_m\boldsymbol{I} \preceq \boldsymbol{A}(\boldsymbol{x}) \preceq t_1\boldsymbol{I} \end{array}$$

in variables $\boldsymbol{x} \in \mathbf{R}^n$ and $t_1, t_m \in \mathbf{R}$.

– Minimize the *spectral radius* (or *spectral norm*) $\rho(\boldsymbol{x}) = \max_{i=1,\ldots,m} |\lambda_i(\boldsymbol{x})|$ is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\boldsymbol{I} \preceq \boldsymbol{A}(\boldsymbol{x}) \preceq t\boldsymbol{I} \end{array}$$

in variables $\boldsymbol{x} \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

– To minimize the condition number $\kappa(\boldsymbol{x}) = \lambda_1(\boldsymbol{x})/\lambda_m(\boldsymbol{x})$, note $\lambda_1(\boldsymbol{x})/\lambda_m(\boldsymbol{x}) \leq t$ if and only if there exists a $\mu > 0$ such that $\mu\boldsymbol{I} \preceq \boldsymbol{A}(\boldsymbol{x}) \preceq \mu t\boldsymbol{I}$, or equivalently, $\boldsymbol{I} \preceq \mu^{-1}\boldsymbol{A}(\boldsymbol{x}) \preceq t\boldsymbol{I}$. With change of variables $y_i = x_i/\mu$ and $s = 1/\mu$, we can solve the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \boldsymbol{I} \preceq s\boldsymbol{A}_0 + y_1\boldsymbol{A}_1 + \cdots y_n\boldsymbol{A}_n \preceq t\boldsymbol{I} \\ & s \geq 0, \end{array}$$

in variables $\boldsymbol{y} \in \mathbf{R}^n$ and $s, t \geq 0$. In other words, we normalize the spectrum by the smallest eigenvalue and then minimize the largest eigenvalue of the normalized LMI.

– Minimize the ℓ_1 norm of the eigenvalues $|\lambda_1(\boldsymbol{x})|+\cdots+|\lambda_m(\boldsymbol{x})|$ is equivalent to the SDP

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(\boldsymbol{A}^+) + \operatorname{tr}(\boldsymbol{A}^-) \\ \text{subject to} & \boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{A}^+ - \boldsymbol{A}^- \\ & \boldsymbol{A}^+ \succeq \boldsymbol{0}, \boldsymbol{A}^- \succeq \boldsymbol{0}, \end{array}$$

in variables $\boldsymbol{x} \in \mathbf{R}^n$ and $\boldsymbol{A}^+, \boldsymbol{A}^- \in \mathbf{S}^n_+$.

– Roots of determinant. The determinant of a semidefinite matrix $\det(\boldsymbol{A}(\boldsymbol{x})) = \prod_{i=1}^m \lambda_i(\boldsymbol{x})$ is neither convex or concave, but rational powers of the determinant can be modeled using linear matrix inequalities. For a rational power $0 \leq q \leq 1/m$, the function $\det(\boldsymbol{A}(\boldsymbol{x}))^q$ is concave and we have

$$\begin{aligned} & t \leq \det(\boldsymbol{A}(\boldsymbol{x}))^q \\ \Leftrightarrow & \begin{pmatrix} \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{Z} \\ \boldsymbol{Z}^T & \operatorname{diag}(\boldsymbol{Z}) \end{pmatrix} \succeq \boldsymbol{0}, \quad (z_{11} z_{22} \cdots z_{mm})^q \geq t, \end{aligned}$$

where $\boldsymbol{Z} \in \mathbf{R}^{m \times m}$ is a lower-triangular matrix. Similarly for any rational $q > 0$, we have

$$\begin{aligned} & t \geq \det(\boldsymbol{A}(\boldsymbol{x}))^{-q} \\ \Leftrightarrow & \begin{pmatrix} \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{Z} \\ \boldsymbol{Z}^T & \operatorname{diag}(\boldsymbol{Z}) \end{pmatrix} \succeq \boldsymbol{0}, \quad (z_{11} z_{22} \cdots z_{mm})^{-q} \leq t \end{aligned}$$

for a lower triangular $\boldsymbol{Z}$.

– Trace of inverse. $\operatorname{tr}\boldsymbol{A}(\boldsymbol{x})^{-1} = \sum_{i=1}^m \lambda_i^{-1}(\boldsymbol{x})$ is a convex function and can be minimized using SDP

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}\boldsymbol{B} \\ \text{subject to} & \begin{pmatrix} \boldsymbol{B} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{A}(\boldsymbol{x}) \end{pmatrix} \succeq \boldsymbol{0}. \end{array}$$

Note $\operatorname{tr}\boldsymbol{A}(\boldsymbol{x})^{-1} = \sum_{i=1}^m \boldsymbol{e}_i^T \boldsymbol{A}(\boldsymbol{x})^{-1} \boldsymbol{e}_i$. Therefore another equivalent formulation is

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^m t_i \\ \text{subject to} & \boldsymbol{e}_i^T \boldsymbol{A}(\boldsymbol{x})^{-1} \boldsymbol{e}_i \leq t_i. \end{array}$$

Now the constraints can be expressed by LMI

$$e_i^T A(x)^{-1} e_i \le t_i \Leftrightarrow \begin{pmatrix} A(x) & e_i \\ e_i^T & t_i \end{pmatrix} \succeq \mathbf{0}.$$

☞ See (Ben-Tal and Nemirovski, 2001, Lecture 4, p146-p151) for the proof of above facts.

☞ `lambda_max`, `lambda_min`, `lambda_sum_largest`, `lambda_sum_smallest`, `det_rootn`, and `trace_inv` are implemented in `cvx` for Matlab.

☞ `lambda_max`, `lambda_min` are implemented in `Convex.jl` package for Julia.

- Example. Experiment design. See HW6 Q1 `http://hua-zhou.github.io/teaching/st790-2015spr/ST790-2015-HW6.pdf`

- Singular value problems. Let $A(x) = A_0 + x_1 A_1 + \cdots x_n A_n$, where $A_i \in \mathbf{R}^{p\times q}$ and $\sigma_1(x) \ge \cdots \sigma_{\min\{p,q\}}(x) \ge 0$ be the ordered singular values.

  - *Spectral norm* (or *operator norm* or *matrix-2 norm*) minimization. Consider minimizing the spectral norm $\|A(x)\|_2 = \sigma_1(x)$. Note $\|A\|_2 \le t$ if and only if $A^T A \preceq t^2 I$ (and $t \ge 0$) if and only if $\begin{pmatrix} tI & A \\ A^T & tI \end{pmatrix} \succeq \mathbf{0}$. This results in the SDP

    $$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{pmatrix} tI & A(x) \\ A(x)^T & tI \end{pmatrix} \succeq \mathbf{0} \end{array}$$

    in variables $x \in \mathbf{R}^n$ and $t \in \mathbf{R}$.

    ☞ Minimizing the sum of k largest singular values is an SDP as well.

  - Nuclear norm minimization. Minimization of the *nuclear norm* (or *trace norm*) $\|A(x)\|_* = \sum_i \sigma_i(x)$ can be formulated as an SDP.

    Argument 1: Singular values of A coincides with the eigenvalues of the symmetric matrix

    $$\begin{pmatrix} \mathbf{0} & A \\ A^T & \mathbf{0} \end{pmatrix},$$

which has eigenvalues $(\sigma_1, \ldots, \sigma_p, -\sigma_p, \ldots, -\sigma_1)$. Therefore minimizing the nuclear norm of $\boldsymbol{A}$ is same as minimizing the ℓ_1 norm of eigenvalues of the augmented matrix, which we know is an SDP.

Argument 2: An alternative characterization of nuclear norm is $\|\boldsymbol{A}\|_* = \sup_{\|\boldsymbol{Z}\|_2 \le 1} \operatorname{tr}(\boldsymbol{A}^T \boldsymbol{Z})$. That is

$$\begin{array}{ll} \text{maximize} & \operatorname{tr}(\boldsymbol{A}^T \boldsymbol{Z}) \\ \text{subject to} & \begin{pmatrix} \boldsymbol{I} & \boldsymbol{Z}^T \\ \boldsymbol{Z} & \boldsymbol{I} \end{pmatrix} \succeq \mathbf{0}, \end{array}$$

with the dual problem

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(\boldsymbol{U} + \boldsymbol{V})/2 \\ \text{subject to} & \begin{pmatrix} \boldsymbol{U} & \boldsymbol{A}(\boldsymbol{x})^T \\ \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{V} \end{pmatrix} \succeq \mathbf{0}. \end{array}$$

Therefore the epigraph of nuclear norm can be represented by LMI

$$\begin{array}{ll} & \|\boldsymbol{A}(\boldsymbol{x})\|_* \le t \\ \Leftrightarrow & \begin{pmatrix} \boldsymbol{U} & \boldsymbol{A}(\boldsymbol{x})^T \\ \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{V} \end{pmatrix} \succeq \mathbf{0}, \quad \operatorname{tr}(\boldsymbol{U} + \boldsymbol{V})/2 \le t. \end{array}$$

Argument 3: See (Ben-Tal and Nemirovski, 2001, Proposition 4.2.2, p154).

☞ See (Ben-Tal and Nemirovski, 2001, Lecture 4, p151-p154) for the proof of above facts.

☞ `sigma_max` and `norm_nuc` are implemented in `cvx` for Matlab.

☞ `operator_norm` and `nuclear_norm` are implemented in `Convex.jl` package for Julia.

- Example. Matrix completion. See HW6 Q2 `http://hua-zhou.github.io/teaching/st790-2015spr/ST790-2015-HW6.pdf`

- Quadratic or quadratic-over-linear matrix inequalities. Suppose

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{x}) &= \boldsymbol{A}_0 + x_1 \boldsymbol{A}_1 + \cdots + x_n \boldsymbol{A}_n \\ \boldsymbol{B}(\boldsymbol{y}) &= \boldsymbol{B}_0 + y_1 \boldsymbol{B}_1 + \cdots + y_r \boldsymbol{B}_r. \end{aligned}$$

Then

$$\begin{aligned} & \boldsymbol{A}(\boldsymbol{x})^T \boldsymbol{B}(\boldsymbol{y})^{-1} \boldsymbol{A}(\boldsymbol{x}) \preceq \boldsymbol{C} \\ \Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{B}(\boldsymbol{y}) & \boldsymbol{A}(\boldsymbol{x})^T \\ \boldsymbol{A}(\boldsymbol{x}) & \boldsymbol{C} \end{pmatrix} \succeq \mathbf{0} \end{aligned}$$

by the Schur complement lemma.

☞ `matrix_frac()` is implemented in both `cvx` for Matlab and `Convex.jl` package for Julia.

- General quadratic matrix inequality. Let $\boldsymbol{X} \in \mathbf{R}^{m \times n}$ be a rectangular matrix and

$$F(\boldsymbol{X}) = (\boldsymbol{AXB})(\boldsymbol{AXB})^T + \boldsymbol{CXD} + (\boldsymbol{CXD})^T + \boldsymbol{E}$$

be a quadratic matrix-valued function. Then

$$\begin{aligned} & F(\boldsymbol{X}) \preceq \boldsymbol{Y} \\ \Leftrightarrow \quad & \begin{pmatrix} \boldsymbol{I} & (\boldsymbol{AXB})^T \\ \boldsymbol{AXB} & \boldsymbol{Y} - \boldsymbol{E} - \boldsymbol{CXD} - (\boldsymbol{CXD})^T \end{pmatrix} \preceq \mathbf{0} \end{aligned}$$

by the Schur complement lemma.

- Another matrix inequality

$$\begin{aligned} & \boldsymbol{X} \succeq \mathbf{0}, \boldsymbol{Y} \preceq (\boldsymbol{C}^T \boldsymbol{X}^{-1} \boldsymbol{C})^{-1} \\ \Leftrightarrow \quad & \boldsymbol{Y} \preceq \boldsymbol{Z}, \boldsymbol{Z} \succeq \mathbf{0}, \boldsymbol{X} \succeq \boldsymbol{CZC}^T. \end{aligned}$$

See (Ben-Tal and Nemirovski, 2001, 20.c, p155).

- Cone of nonnegative polynomials. Consider nonnegative polynomial of degree $2n$

$$f(t) = \boldsymbol{x}^T \boldsymbol{v}(t) = x_0 + x_1 t + \cdots x_{2n} t^{2n} \geq 0, \text{ for all } t.$$

The cone

$$\mathbf{K}^n = \{\boldsymbol{x} \in \mathbf{R}^{2n+1} : f(t) = \boldsymbol{x}^T \boldsymbol{v}(t) \geq 0, \text{ for all } t \in \mathbf{R}\}$$

can be characterized by LMI

$$f(t) \geq 0 \text{ for all } t \Leftrightarrow x_i = \langle \boldsymbol{X}, \boldsymbol{H}_i \rangle, i = 0, \ldots, 2n, \boldsymbol{X} \in \mathbf{S}_+^{n+1},$$

where $\boldsymbol{H}_i \in \mathbf{R}^{(n+1)\times(n+1)}$ are Hankel matrices with entries $(\boldsymbol{H}_i)_{kl} = 1$ if $k+l = i$ or 0 otherwise. Here $k, l \in \{0, 1, \ldots, n\}$.

Similarly the cone of nonnegative polynomials on a finite interval

$$\mathbf{K}_{a,b}^n = \{\boldsymbol{x} \in \mathbf{R}^{n+1} : f(t) = \boldsymbol{x}^T \boldsymbol{v}(t) \geq 0, \text{ for all } t \in [a, b]\}$$

can be characterized by LMI as well.

- (Even degree) Let $n = 2m$. Then

$$\begin{aligned} \mathbf{K}_{a,b}^n &= \{\boldsymbol{x} \in \mathbf{R}^{n+1} : x_i = \langle \boldsymbol{X}_1, \boldsymbol{H}_i^m \rangle + \langle \boldsymbol{X}_2, (a+b)\boldsymbol{H}_{i-1}^{m-1} - ab\boldsymbol{H}_i^{m-1} - \boldsymbol{H}_{i-2}^{m-1} \rangle, \\ &\quad i = 0, \ldots, n, \boldsymbol{X}_1 \in \mathbf{S}_+^m, \boldsymbol{X}_2 \in \mathbf{S}_+^{m-1}\}. \end{aligned}$$

- (Odd degree) Let $n = 2m + 1$. Then

$$\begin{aligned} \mathbf{K}_{a,b}^n &= \{\boldsymbol{x} \in \mathbf{R}^{n+1} : x_i = \langle \boldsymbol{X}_1, \boldsymbol{H}_{i-1}^m - a\boldsymbol{H}_i^m \rangle + \langle \boldsymbol{X}_2, b\boldsymbol{H}_i^m - \boldsymbol{H}_{i-1}^m \rangle, \\ &\quad i = 0, \ldots, n, \boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbf{S}_+^m\}. \end{aligned}$$

☞ References: paper (Nesterov, 2000) and the book (Ben-Tal and Nemirovski, 2001, Lecture 4, p157-p159).

- Example. Polynomial curve fitting. We want to fit a univariate polynomial of degree n

$$f(t) = x_0 + x_1 t + x_2 t^2 + \cdots x_n t^n$$

to a set of measurements (t_i, y_i), $i = 1, \ldots, m$, such that $f(t_i) \approx y_i$. Define the Vandermonde matrix

$$\boldsymbol{A} = \begin{pmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^n \\ 1 & t_2 & t_2^2 & \cdots & t_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^n \end{pmatrix},$$

then we wish $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{y}$. Using least squares criterion, we obtain the optimal solution $\boldsymbol{x}_{\text{LS}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$. With various constraints, it is possible to find optimal $\boldsymbol{x}$ by SDP.

1. Nonnegativity. Then we require $\boldsymbol{x} \in \mathbf{K}_{a,b}^n$.
2. Monotonicity. We can ensure monotonicity of $f(t)$ by requiring that $f'(t) \geq 0$ or $f'(t) \leq 0$. That is $(x_1, 2x_2, \ldots, nx_n) \in \mathbf{K}_{a,b}^{n-1}$ or $-(x_1, 2x_2, \ldots, nx_n) \in \mathbf{K}_{a,b}^{n-1}$.
3. Convexity or concavity. Convexity or concavity of $f(t)$ corresponds to $f''(t) \geq 0$ or $f''(t) \leq 0$. That is $(2x_2, 2x_3, \ldots, (n-1)nx_n) \in \mathbf{K}_{a,b}^{n-2}$ or $-(2x_2, 2x_3, \ldots, (n-1)nx_n) \in \mathbf{K}_{a,b}^{n-2}$.

☞ `nonneg_poly_coeffs()` and `convex_poly_coeffs()` are implemented in `cvx`. Not in `Convex.jl` yet.

- SDP relaxation of binary optimization. Consider a binary linear optimization problem

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \in \{0,1\}^n. \end{array}$$

Note

$$x_i \in \{0,1\} \Leftrightarrow x_i^2 = x_i \Leftrightarrow \boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^T, \operatorname{diag}(\boldsymbol{X}) = \boldsymbol{x}.$$

By relaxing the rank 1 constraint on $\boldsymbol{X}$, we obtain an SDP relaxation

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \operatorname{diag}(\boldsymbol{X}) = \boldsymbol{x}, \boldsymbol{X} \succeq \boldsymbol{x}\boldsymbol{x}^T, \end{array}$$

which can be efficiently solved and provides a lower bound to the original problem. If the optimal $\boldsymbol{X}$ has rank 1, then it is a solution to the original binary problem also. Note $\boldsymbol{X} \succeq \boldsymbol{x}\boldsymbol{x}^T$ is equivalent to the LMI

$$\begin{pmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \succeq \mathbf{0}.$$

We can tighten the relaxation by adding other constraints that cut away part of the feasible set, without excluding rank 1 solutions. For instance, $0 \leq x_i \leq 1$ and $0 \leq X_{ij} \leq 1$.

- SDP relaxation of boolean optimization. For Boolean constraints $\boldsymbol{x} \in \{-1,1\}^n$, we note

$$x_i \in \{0,1\} \Leftrightarrow \boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^T, \operatorname{diag}(\boldsymbol{X}) = \mathbf{1}.$$

Geometric programming (GP)

- A function $f : \mathbf{R}^n \mapsto \mathbf{R}$ with $\text{dom} f = \mathbf{R}^n_{++}$ defined as

$$f(\boldsymbol{x}) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n},$$

  where $c > 0$ and $a_i \in \mathbf{R}$, is called a *monomial*.

- A sum of monomials

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}},$$

  where $c_k > 0$, is called a *posynomial*.

- Posynomials are closed under addition, multiplication, and nonnegative scaling.

- A *geometric program* is of form

$$\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_i(\boldsymbol{x}) \leq 1, \quad i = 1, \ldots, m \\ & h_i(\boldsymbol{x}) = 1, \quad i = 1, \ldots, p \end{array}$$

  where $f_0, \ldots, f_m$ are posynomials and $h_1, \ldots, h_p$ are monomials. The constraint $\boldsymbol{x} \succ \mathbf{0}$ is implicit.

  ☞ Is GP a convex optimization problem?

- With change of variable $y_i = \ln x_i$, a monomial

$$f(\boldsymbol{x}) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

  can be written as

$$f(\boldsymbol{x}) = f(e^{y_1}, \ldots, e^{y_n}) = c(e^{y_1})^{a_1} \cdots (e^{y_n})^{a_n} = e^{\boldsymbol{a}^T \boldsymbol{y} + b},$$

  where $b = \ln c$. Similarly, we can write a posynomial as

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}} \\ &= \sum_{k=1}^{K} e^{\boldsymbol{a}_k^T \boldsymbol{y} + b_k}, \end{aligned}$$

  where $\boldsymbol{a}_k = (a_{1k}, \ldots, a_{nk})$ and $b_k = \ln c_k$.

- The original GP can be expressed in terms of the new variable $\boldsymbol{y}$

$$\begin{array}{ll} \text{minimize} & \sum_{k=1}^{K_0} e^{\boldsymbol{a}_{0k}^T \boldsymbol{y} + b_{0k}} \\ \text{subject to} & \sum_{k=1}^{K_i} e^{\boldsymbol{a}_{ik}^T \boldsymbol{y} + b_{ik}} \le 1, \quad i = 1, \ldots, m \\ & e^{\boldsymbol{g}_i^T \boldsymbol{y} + h_i} = 1, \quad i = 1, \ldots, p, \end{array}$$

where $\boldsymbol{a}_{ik}, \boldsymbol{g}_i \in \mathbf{R}^n$. Taking log of both objective and constraint functions, we obtain the *geometric program in convex form*

$$\begin{array}{ll} \text{minimize} & \ln \left(\sum_{k=1}^{K_0} e^{\boldsymbol{a}_{0k}^T \boldsymbol{y} + b_{0k}} \right) \\ \text{subject to} & \ln \left(\sum_{k=1}^{K_i} e^{\boldsymbol{a}_{ik}^T \boldsymbol{y} + b_{ik}} \right) \le 0, \quad i = 1, \ldots, m \\ & \boldsymbol{g}_i^T \boldsymbol{y} + h_i = 0, \quad i = 1, \ldots, p. \end{array}$$

☞ Mosek is capable of solving GP. `cvx` has a GP mode that recognizes and transforms GP problems.

- Example. Logistic regression as GP. Given data $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, where $y_i \in \{0, 1\}$ and $\boldsymbol{x}_i \in \mathbf{R}^p$, the likelihood of the logistic regression model is

$$\begin{aligned} & \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1 - y_i} \\ = & \prod_{i=1}^n \left(\frac{e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} \right)^{y_i} \left(\frac{1}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} \right)^{1 - y_i} \\ = & \prod_{i: y_i = 1} e^{\boldsymbol{x}_i^T \boldsymbol{\beta} y_i} \prod_{i=1}^n \left(\frac{1}{1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}}} \right). \end{aligned}$$

The MLE solves

$$\text{minimize} \quad \prod_{i: y_i = 1} e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} \prod_{i=1}^n \left(1 + e^{\boldsymbol{x}_i^T \boldsymbol{\beta}} \right).$$

Let $z_j = e^{\beta_j}$, $j = 1, \ldots, p$. The objective becomes

$$\prod_{i: y_i = 1} \prod_{j=1}^p z_j^{-x_{ij}} \prod_{i=1}^n \left(1 + \prod_{j=1}^p z_j^{x_{ij}} \right).$$

This leads to a GP

$$
\begin{aligned}
\text{minimize} \quad & \prod_{i:y_i=1} s_i \prod_{i=1}^{n} t_i \\
\text{subject to} \quad & \prod_{j=1}^{p} z_j^{-x_{ij}} \le s_i, \quad i = 1, \ldots, m \\
& 1 + \prod_{j=1}^{p} z_j^{x_{ij}} \le t_i, \quad i = 1, \ldots, n,
\end{aligned}
$$

in variables $\boldsymbol{s} \in \mathbf{R}^m$, $\boldsymbol{t} \in \mathbf{R}^n$, and $\boldsymbol{z} \in \mathbf{R}^p$. Here m is the number of observations with $y_i = 1$.

☞ How to incorporate lasso penalty? Let $z_j^+ = e^{\beta_j^+}$, $z_j^- = e^{\beta_j^-}$. Lasso penalty takes the form $e^{\lambda|\beta_j|} = (z_j^+ z_j^-)^\lambda$.

- Example. Bradley-Terry model for sports ranking. See ST758 HW8 `http://hua-zhou.github.io/teaching/st758-2014fall/ST758-2014-HW8.pdf`. The likelihood is

  $$\prod_{i,j} \left(\frac{\gamma_i}{\gamma_i + \gamma_j} \right)^{y_{ij}}.$$

  MLE is solved by GP

  $$
  \begin{aligned}
  \text{minimize} \quad & \prod_{i,j} t_{ij}^{y_{ij}} \\
  \text{subject to} \quad & 1 + \gamma_i^{-1} \gamma_j \le t_{ij}
  \end{aligned}
  $$

  in $\boldsymbol{\gamma} \in \mathbf{R}^n$ and $\boldsymbol{t} \in \mathbf{R}^{n^2}$.

Generalized inequalities and cone programming

- A cone $\mathbf{K} \in \mathbf{R}^n$ is *proper* if it is closed, convex, has non-empty interior, and is pointed, i.e., $\boldsymbol{x} \in \mathbf{K}$ and $-\boldsymbol{x} \in \mathbf{K}$ implies $\boldsymbol{x} = \mathbf{0}$.

  A proper cone defines a partial ordering on $\mathbf{R}^n$ via *generalized inequalities*: $\boldsymbol{x} \preceq_{\mathbf{K}} \boldsymbol{y}$ if and only if $\boldsymbol{y} - \boldsymbol{x} \in \mathbf{K}$ and $\boldsymbol{x} \prec \boldsymbol{y}$ if and only if $\boldsymbol{y} - \boldsymbol{x} \in \text{int}(\mathbf{K})$.

  E.g., $\boldsymbol{X} \preceq \boldsymbol{Y}$ means $\boldsymbol{Y} - \boldsymbol{X} \in \mathbf{S}_+^n$ and $\boldsymbol{X} \prec \boldsymbol{Y}$ means $\boldsymbol{Y} - \boldsymbol{X} \in \mathbf{S}_{++}^n$.

- A *conic form problem* or *cone program* has the form

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{F}\boldsymbol{x}+\boldsymbol{g} \preceq_K \boldsymbol{0} \\ & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}. \end{array}$$

- The *conic form problem in standard form* is

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \succeq_K \boldsymbol{0} \\ & \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}. \end{array}$$

- The *conic form problem in inequality form* is

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{F}\boldsymbol{x}+\boldsymbol{g} \preceq_K \boldsymbol{0}. \end{array}$$

- Special cases of cone programming.
  - Nonnegative orthant $\{\boldsymbol{x}|\boldsymbol{x} \succeq \boldsymbol{0}\}$: LP
  - Second order cone $\{(\boldsymbol{x},t)|\|\boldsymbol{x}\|_2 \le t\}$: SOCP
  - Rotated quadratic cone $\{(\boldsymbol{x},t_1,t_2)|\|\boldsymbol{x}\|_2^2 \le 2t_1t_2\}$: SOCP
  - Geometric mean cone $\{(\boldsymbol{x},t)|(\prod x_i)^{1/n} \ge y, \boldsymbol{x} \succeq \boldsymbol{0}\}$: SOCP
  - Semidefinite cone $\mathbf{S}_+^n$: SDP
  - Nonnegative polynomial cone: SDP
  - Monotone polynomial cone: SDP
  - Convex/concave polynomial cone: SDP
  - Exponential cone $\{(x,y,z)|ye^{x/y} \le z, y>0\}$. Terms `logsumexp`, `exp`, `log`, `entropy`, `lndet`, ... are exponential cone representable.

- Where is today's technology up to?
  - `Gurobi` implements up to SOCP.
  - `Mosek` implements up to SDP.

- `SCS` (free solver accessible from `Convex.jl`) can deal with exponential cone program.
- `cvx` uses a successive approximation strategy to deal with exponential cone representable terms, which only relies on SOCP.
  `http://web.cvxr.com/cvx/doc/advanced.html#successive`
  ☞ `cvx` implements `log_det` and `log_sum_exp`.
- `Convex.jl` accepts exponential cone representable terms, which can solve using SCS.
  ☞ `Convex.jl` implements `logsumexp`, `exp`, `log`, `entropy`, and `logistic_loss`.

• Example. Logistic regression as an exponential cone problem

$$\text{minimize} \quad -\sum_{i:y_i=1} \boldsymbol{x}_i^T\boldsymbol{\beta} + \sum_{i=1}^{n} \ln\left(1 + e^{\boldsymbol{x}_i^T\boldsymbol{\beta}}\right).$$

See `cvx` example library for an example for logistic regression. `http://cvxr.com/cvx/examples/`
See the link for an example using Julia. `http://nbviewer.ipython.org/github/JuliaOpt/Convex.jl/blob/master/examples/logistic_regression.ipynb`

• Example. Gaussian covariance estimation and graphical lasso

$$\ln\det(\boldsymbol{\Sigma}) + \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Sigma}) - \lambda\|\text{vec}\Sigma\|_1$$

involves exponential cones because of the $\ln\det$ term.

Separable convex optimization in Mosek

• Mosek is posed to solve general convex nonlinear programs (NLP) of form

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) + \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & l_i \le g_i(\boldsymbol{x}) + \boldsymbol{a}_i^T\boldsymbol{x} \le u_i, \quad i = 1, \ldots, m \\ & \boldsymbol{l}^x \preceq \boldsymbol{x} \preceq \boldsymbol{u}^x. \end{array}$$

Here functions $f : \mathbf{R}^n \mapsto \mathbf{R}$ and $g_i : \mathbf{R}^n \mapsto \mathbf{R}$, $i = 1, \ldots, m$ must be *separable* in parameters.

- The example

$$\begin{aligned} \text{minimize} \quad & x_1 - \ln(x_1 + 2x_2) \\ \text{subject to} \quad & x_1^2 + x_2^2 \le 1 \end{aligned}$$

is not separable. But the equivalent formulation

$$\begin{aligned} \text{minimize} \quad & x_1 - \ln(x_3) \\ \text{subject to} \quad & x_1^2 + x_2^2 \le 1, x_1 + 2x_2 - x_3 = 0, x_3 \ge 0 \end{aligned}$$

is.

- It should cover a lot statistical applications. But I have no experience with its performance yet.

- Which modeling tool to use?
  - `cvx` and `Convex.jl` can *not* model general NLP.
  - `JuMP.jl` in `Julia` can model NLP or even MINLP. See `http://jump.readthedocs.org/en/latest/nlp.html`

Other topics in convex optimization

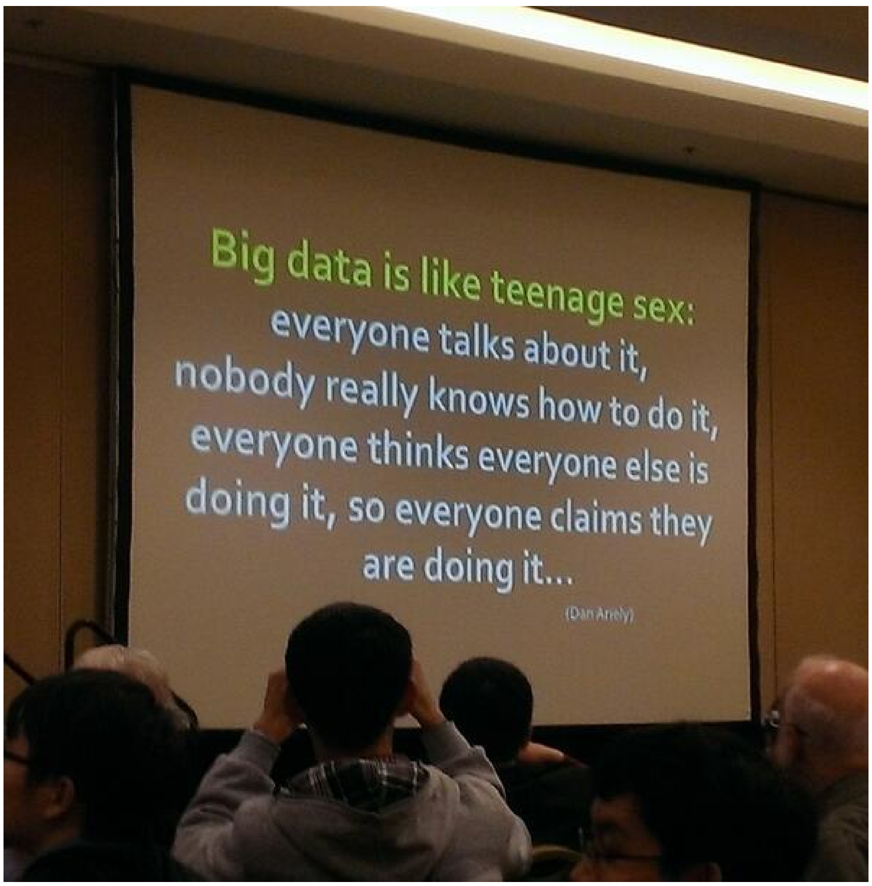
- Duality theory. (Boyd and Vandenberghe, 2004, Chapter 5).

- Algorithms. Interior point method. (Boyd and Vandenberghe, 2004) Part III (Chapters 9-11).

- History:
  1. 1948: Dantzig's simplex algorithm for solving LP.
  2. 1950s: many applications of LP in operations research, network optimization, finance, engineering, ...
  3. 1950s: quadratic programming (QP).
  4. 1960s: geometric programming (GP).
  5. 1984: first practical polynomial-time algorithm (interior point method) by Karmarkar.

6. 1984-1990: efficient implementations for large-scale LP.
7. around 1990: polynomial-time interior-point methods for nonlinear convex programming by Nesterov and Nemirovski.
8. since 1990: extensions (QCQP, SOCP, SDP) and high-quality software packages.

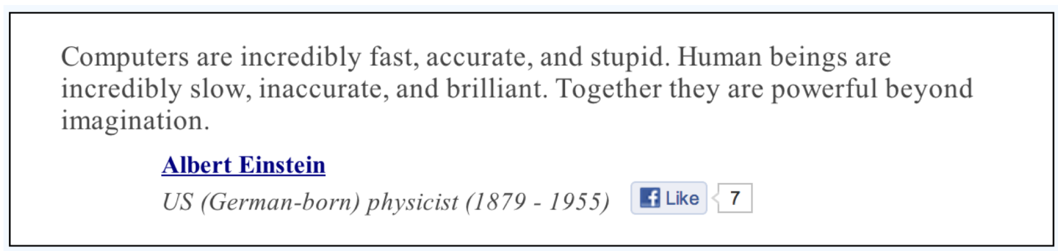
Take-home messages from this course



- Statistics, the science of *data analysis*, is the applied mathematics in the 21st century
  - Read the the article *50 Years of Data Science* by David Donoho.
- *Big data* era: Challenges also mean opportunities for statisticians
  - methodology: big p

  - efficiency: big n and/or big p
  - memory: big n, distributed computing via MapReduce (Hadoop), online algorithms

- Being good at computing (*both* programming and algorithms) is a must for today's working (bio)statisticians.


Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination.

Albert Einstein

US (German-born) physicist (1879 - 1955)

- HPC (high performance computing) $\neq$ abusing computers.

  Always optimize your algorithms *as much as possible* before resorting to cluster computing resources. In this course we see many examples where careful algorithm choice and coding yields $>$ 10-fold or even $>$ 100-fold speedup.

- Coding
  - Prototyping: Julia, Matlab, R
  - A "real" programming language: Julia, C/C++, Fortran, Python
  - Scripting language: Python, Linux/Unix script, Perl, JavaScript
  - Be reproducible: git and dynamic document

- Numerical linear algebra – building blocks of most computing we do. Use standard *libraries* (BLAS, LAPACK, ...)! Sparse linear algebra and iterative solvers such as conjugate gradient (CG) methods are critical for exploiting structure in big data.

- Optimization
  - *Convex programming* (LS, LP, QP, GP, SOCP, SDP). Download and study Stephen Boyd's book, watch lecture vides or take EE236B (*Convex Optimization* taught by Vandenberghe), familiarize yourself with the *good* optimization softwares. Convex programming is becoming a *technology*, just like least squares (LS).

  - Generic nonlinear optimization tools: Newton, Gauss-Newton, quasi-Newton, (nonlinear) conjugate gradient, ...
  - Optimization tools developed by statisticians: Fisher scoring, EM, MM, ...

- Culture: know the names. John Tukey (FFT, box-plot, bit, multiple testing, ...), David Donoho (wavelet, lasso, reproducible research, ...), Stephen Boyd, Lieven Vandenberghe, Nesterov, Nemirovski, Kenneth Lange, Hadley Wickham, Dantzig, ...

- Things I didn't do in this class:
  - MCMC: take a Bayesian course!
  - Specialized optimization algorithms for large scale statistical learning problems: coordinate descent, proximal gradient (with Nesterov acceleration), ALM, ADMM, ... Take EE236C (*Optimization Methods for Large-scale Systems* taught by Vandenberghe).
  - Combinatorial optimization techniques: divide-and-conquer, dynamic programming (e.g., HMM), greedy algorithm, simulated annealing, ...