CVX tutorial session

TA: Mykola Servetnyk

Instructor: Professor Carrson C. Fung

Optimization Theory and Applications

National Chiao Tung University

11 April, 2019

What is CVX?

- Website: http://cvxr.com/cvx
- CVX is a modeling system for disciplined convex programming (DCP)
  - http://cvxr.com/cvx/doc/dcp.html
- DCPs are convex optimization problems that are described using a limited set of construction rules, which enables them to be analyzed and solved efficiently
- CVX is implemented in Matlab
- CVX relies on external solvers to solve problems
  - Mosek, Gurobi, SDPT3, SeDuMi
- If CVX accepts your problem, you can be sure it is convex
  - Reverse is not always true: Problems that violate the ruleset are rejected, even when the problem is convex

Mathematically

$$
\begin{aligned}
\min\ & f_0(x), i = 1, ..., m \\
\text{s.t. } & f_i(x) \leq 0, i = 1, ..., p \\
& h_i(x) = 0
\end{aligned}
$$

In CVX

```
cvx_begin
  variables x(n)
  minimize(f0(x))
  subject to
    f(x) <= 0
    A * x - b == 0
cvx_end
```

Return variables

Upon exit, CVX sets the variables

- `x`-solution variable x^*
- `cvx_optval` - solution variable x^*
- `cvx_status` - solver status (`Solved`, `Unbounded`, `Infeasible`,...)
- ...

Examples - Basic

Optimization problem

$$\begin{aligned} \min\ & x + y \\ \text{s.t. } & x \geq 1 \\ & y = 2 \end{aligned}$$

In CVX

```
cvx_begin
  variables x(1) y(1)
  minimize (x+y)
  subject to
    x >= 1
    y == 2
cvx_end
```

Examples - Basic

CVX returns a solution and status

```
>> x
    1
>> y
    2
>> cvx_optval
    3
>> cvx_status
Solved
```

Examples - LP

Optimization problem

$$\max \mathbf{c}^T \mathbf{x}$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \succeq \mathbf{0}$$

In CVX

```
cvx_begin
  variable x(n)
  maximize (c' * x)
  subject to
    A * x  == b
    x >= 0
cvx_end
```

Examples - SDP

Optimization problem

$$\min \|\mathbf{X} - \mathbf{A}\|_2^2$$
$$\text{s.t. } \mathbf{X} \succeq \mathbf{0}$$

In CVX

```
cvx_begin
  variable X(n,n) semidefinite
  minimize square_pos(norm(A - X))
  subject to
    X == semidefinite(n)
cvx_end
```

Examples - Assignments

Optimization problem

$$\min \sum_i \|\mathbf{x} - \mathbf{a}_i\|_2$$

In CVX

```
cvx_begin
  variable x(n,1)
  OBJ = cvx(0);
  for i = 1:n
     OBJ = OBJ + norm(x-A(:,i));
  end
  minimize norm(OBJ)
cvx_end
```

CVX - Variables

The currently supported structure keywords are

- `banded(lb,ub)`
- `diagonal`
- `hankel`
- `hermitian`
- `skew_symmetric`
- `symmetric`
- `toeplitz`
- `tridiagonal`
- `lower_bidiagonal`
- `lower_hessenberg`
- `lower_triangular`
- `upper_bidiagonal`
- `upper_hankel`
- `upper_hessenberg`
- `upper_triangular`

Supported properties

- `complex`
- `nonnegative`
- `semidefinite`
- `integer`
- `binary`

Some functions

function	meaning	attributes
`norm(x, p)`	$\|x\|_p$	cvx
`square(x)`	x^2	cvx
`square_pos(x)`	$(x_+)^2$	cvx, nondecr
`pos(x)`	x_+	cvx, nondecr
`sum_largest(x,k)`	$x_{[1]} + \cdots + x_{[k]}$	cvx, nondecr
`sqrt(x)`	$\sqrt{x} \quad (x \geq 0)$	ccv, nondecr
`inv_pos(x)`	$1/x \quad (x > 0)$	cvx, nonincr
`max(x)`	$\max\{x_1, \ldots, x_n\}$	cvx, nondecr
`quad_over_lin(x,y)`	$x^2/y \quad (y > 0)$	cvx, nonincr in y
`lambda_max(X)`	$\lambda_{\max}(X) \quad (X = X^T)$	cvx
`huber(x)`	$\begin{cases} x^2, & \|x\| \leq 1 \\ 2\|x\| - 1, & \|x\| > 1 \end{cases}$	cvx

Composition rules

It is possible to form composite functions

- a convex function of an ane function is convex
- the negative of a convex function is concave
- a convex, nondecreasing function of a convex function is convex
- a concave, nondecreasing function of a concave function is concave

- **Objective** can be
  - `minimize(convex expression)`
  - `minimize(convex expression)`
  - omitted (feasibility problem)
- **Constraint** can be
  - `convex expression <= concave expression`
  - `concave expression <= convex expression`
  - `affine expression == affine expression`
  - omitted (unconstrained problem)

CVX hints/warnings

- watch out for = (assignment) versus == (equality constraint)
- `X>=0`, with matrix `X` is an elementwise inequality
- writing `subject to` is unnecessary (but can look nicer)
- many problems traditionally stated using convex quadratic forms can posed as norm problems

  `x'*P*x <= 1` can be replaced with `norm(chol(P)*x) <= 1`

Reference

- CVX user's guide: http://web.cvxr.com/cvx/beta/doc/CVX.pdf

Remark

- Some problems do not appear to be convex
- Even if it is known that problem is convex, it might require some reformulations to program it to cvx (or make it run much faster)
- It may help you to understand how cvx works and do your own problem reformulations.

Example: Chebyshev (ℓ_∞) norm

$$\text{minimize} \quad f(Ax-b) \triangleq \|Ax-b\|_\infty = \max_{1\le k\le m} |a_k^T x - b|$$

This can be expressed as a linear program:

$$\begin{array}{ll} \text{minimize} & q \\ \text{subject to} & -q\mathbf{1} \le Ax - b \le +q\mathbf{1} \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} A & -\mathbf{1} \\ -A & -\mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} \le \begin{bmatrix} b \\ -b \end{bmatrix} \end{array}$$

Example: Manhattan (ℓ_1) norm

$$\text{minimize} \quad f(Ax-b) \triangleq \|Ax-b\|_1 = \sum_{k=1}^{m} |a_k^T x - b|$$

LP formulation:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T w \\ \text{subject to} & -w \le Ax - b \le w \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}^T \begin{bmatrix} x \\ w \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \le \begin{bmatrix} b \\ -b \end{bmatrix} \end{array}$$

Example: Largest-L norm

Let $|w|_{[k]}$ represent the k-th largest element (in magnitude) of $w \in \mathbf{R}^m$:

$$|w|_{[1]} \geq |w|_{[2]} \geq \cdots \geq |w|_{[m]}$$

Then define the *largest-L norm* as

$$\|w\|_{[L]} \triangleq |w|_{[1]} + |w|_{[2]} + \cdots + |w|_{[L]} \triangleq \textstyle\sum_{k=1}^{L} \sigma_k(\mathbf{diag}(w))$$

for any integer $1 \leq L \leq m$. Special cases include

$$\|w\|_{[1]} \equiv \|w\|_\infty, \quad \|w\|_{[m]} \triangleq \|w\|_1$$

but novel results are produced for all $1 < L < m$

Example: Constrained ℓ_2 norm

Add some constraints:

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2 \\ \text{subject to} & Cx = d \\ & \ell \le x \le u \end{array}$$

This is not a least-squares problem—but it is an SOCP, and SeDuMi can handle it, once it is converted to standard form:

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & Ax - b = y \\ & Cx = d \\ & x - s_\ell = \ell \\ & x + s_u = u \\ & s_\ell, s_u \ge 0 \\ & \|y\|_2 \le z \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}^T \bar{x} \\ \text{subject to} & \begin{bmatrix} A & & & -I & \\ C & & & & \\ I & -I & & & \\ I & & I & & \end{bmatrix} \bar{x} = \begin{bmatrix} b \\ d \\ \ell \\ u \end{bmatrix} \\ & \bar{x} \in \mathbf{R}^n \times \mathbf{R}^n_+ \times \mathbf{R}^n_+ \times \mathbf{Q}^m \end{array}$$

$s_l, s_u \in \mathbf{R}^n_+$ are *slack variables*, which are used quite often to convert inequalities to equations

Good Luck!