CVX tutorial session

EE364A

Stanford University

January 22, 2015

What is CVX?

- CVX is a modeling system for convex optimization problems
- Website: `http://cvxr.com/cvx`

Installation on Corn

In terminal

```
$ ssh -Y SUNETID@corn.stanford.edu
$ wget http://web.cvxr.com/cvx/cvx-a64.tar.gz
$ tar -xvf cvx-a64.tar.gz
$ module load matlab
$ matlab &
```

In MATLAB

```
>> cd cvx
>> cvx_setup
```

Structure of convex problem

Mathematically

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p\end{array}$$

In CVX

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

NB: f_0 and f_i must be convex and h_i must be affine

Return values

Upon exit, CVX sets the variables

- `x` – solution variable(s) $x^\star$
- `cvx_optval` – the optimal value $p^\star$
- `cvx_status` – solver status (`Solved`, `Unbounded`, `Infeasible`, ..)
- ...

Examples – Basic

Optimization problem

$$\begin{array}{ll} \text{minimize} & x + y \\ \text{subject to} & x \geq 1, \quad y = 2. \end{array}$$

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

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax = b \\ & x \succeq 0 \end{array}$$

In CVX:

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

Examples – SDP

Optimization problem

$$\begin{array}{ll} \text{minimize} & \|X - A\|_2 \\ \text{subject to} & X \succeq 0 \end{array}$$

In CVX:

```
cvx_begin sdp
   variable X(n, n)
   minimize(norm(A - X))
   subject to
      X >= 0
cvx_end
```

Examples – Assignments

Optimization problem

$$\text{minimize} \quad \sum_i \|x - a_i\|_2$$

In CVX:

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

NB: This can be *really slow* for large loops!

CVX Pitfalls

- Use == for equality constraints, not =
- Use >= and <= instead of > and <
- Each functions must be convex on its domain (not just in feasible region).

Resources

- CVX documentation http://web.cvxr.com/cvx/doc/
- Disciplined convex optimization http://dcp.stanford.edu/

Questions?