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

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad h_i(x) = 0, \quad i = 1, \ldots, p
\end{align*}
\]

In CVX

```cvx
variables x(n)
minimize(f0(x))
snbject to
  f(x) <= 0
  A * x - b == 0
```

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) $x^*$
- `cvx_optval` – the optimal value $p^*$
- `cvx_status` – solver status (Solved, Unbounded, Infeasible, ..)
- ...

Convex problems
Examples – Basic

Optimization problem

\[
\begin{align*}
\text{minimize} & \quad x + y \\
\text{subject to} & \quad x \geq 1, \quad y = 2.
\end{align*}
\]

In CVX:

```
cvx_begin
  variables x(1) y(1)
  minimize(x + y)
  subject to
    x >= 1
    y == 2
cvx_end
```
CVX returns a solution and status

```matlab
>> x
    1

>> y
    2

>> cvx_optval
    3

>> cvx_status
Solved
```
Examples – LP

Optimization problem

\[
\begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \\
& \quad x \succeq 0
\end{align*}
\]

In CVX:

```plaintext
cvx_begin
variables x(n)
maximize(c' * x)
subject to
A * x == b
x >= 0
cvx_end
```
Examples – SDP

Optimization problem

\begin{align*}
\text{minimize} & \quad \|X - A\|_2 \\
\text{subject to} & \quad X \succeq 0
\end{align*}

In CVX:

\begin{verbatim}
cvx_begin sdp
variable X(n, n)
minimize(norm(A - X))
subject to
    X >= 0
cvx_end
\end{verbatim}
Examples – Assignments

Optimization problem

\[
\text{minimize } \sum_i \|x - a_i\|_2
\]

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 function 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?