CS 486/686 – Matlab Tutorial

Introduction to Matlab with
a focus on Assignment 4

David L. Yeung

(for Prof. Pascal Poupart)

What is Matlab?

- "Matlab" comes from the words Matrix Laboratory:
  - It's an environment for scientific computing based on *matrix manipulation*.
  - It has an interactive environment, a function library, and a programming language.

Starting and Exiting Matlab

- To start Matlab:
  - Type "`matlab`" at the Unix prompt.
  - Options: "`matlab –nojvm –nosplash`"
- To exit Matlab:
  - Type "`exit`" or "`quit`" at the Matlab prompt.

Getting HELP

- The most important command:
  - Type "`help`" at the Matlab prompt.
  - Type "`help command`" for help on a command.
- Other useful help commands:
  - "`lookfor`" – search for a command.
  - "`helpdesk`" or "`helpwin`" – help window.

Matrices

- (Almost) everything is a matrix:
  - `> A = [1 2; 3 4]`
    ```
    A =
            1    2
            3    4
    ```
  - `> A(2,1)`
    ```
    ans =
            3
    ```
- Can also have higher dimensions:
  - `> T = zeros(17,17,4);` (note semicolon)
  - `> size(T)`
    ```
    ans =
          17   17   4
    ```

Matrices (cont'd)

- Initialization:
  - zeros, ones, eye, cat
  - `> Uprime = zeros(17,1);`
- Submatrices:
  - `> A(2,:)`
    ```
    ans =
            3   4
    ```
  - `> A(:,1)`
    ```
    ans =
            1
            3
    ```

Matrix manipulation

- Operate on entire matrices:
  - Add, subtract, multiply, divide.
  - > `C = A + B;`
- Operate on each element:
  - > `C = A .* B;`
  - See “`help times`”, “`help mtimes`”.

Helpful functions

- For Assignment 4, you might want to take a look at the "`help`" for:
  - `max` (`min`), `abs`, `sum`
    - `delta = max(delta, abs(Uprime(s) - U(s)));`
- Suppose we want (for a fixed s):

$$\max_a \sum_{s'} T(s, s', a) U(s')$$

Helpful functions - Example

- First, we look at:

$$T(s, s', a)U(s')$$

- Recall that the *size* of *T* is (17,17,4):
  - Thus, the *size* of *T(s,:,:)* is (1,17,4).
  - But the *size* of *U* is (17,1).
- To multiply *T* and *U* on index *s'*, first:
  - `U4 = repmat(U,1,4) % same as [U U U U]`
  - which now has a *size* of (17,4).

Helpful functions - Example

- We want:

$$T(s, s', a)U(s')$$

- We define:
  - `TU = squeeze(T(s,:,:)) .* U4;`
  - See "`help squeeze`".
- Then: $\max_a \sum_{s'} T(s, s', a)U(s')$

  is just: `max(sum(TU));`

Loops

- Matlab has both “for” and “while”:

```
 i = 1;
for t = 0:pi/20:pi,
    y(i) = sin(t);
    i = i+1;
end
```

- “For” loops can often be replaced:

```
t = 0: pi/20 : pi;
y = sin(t)
```

- See help for “`for`”, “`while`”, “`break`”.

Scripts and Functions

- Matlab can execute ".m" files:
  - Can either be scripts or functions.
- Function definitions:
  - Result, function name, input args.
    - `function [T, R] = gridWorld`
    - `function utility = valueIteration(T, R, gamma, epsilon)`
    - `function showGrid(U)`
- Try changing to:
    - `function [T,R] = gridWorld(a,b)`

Other useful commands

- Workspace:
  - `whos`, `size`
- Matrix manipulation:
  - `find`, `relop`, `logical`
- Display:
  - `disp`, `sprintf`, `(;)`
- Diary:
  - `diary on`, `diary off`, `diary('file')`

Value Iteration

```
function U = valueIteration(T, R, gamma, epsilon)
    Uprime = ...
    while true, % repeat
        U = Uprime; delta = 0;

        for s = 1:17, % for each state s in S do
            Uprime(s) = ... % recall TU from earlier

            delta = ... % update delta
        end % for

        if delta ... % do until condition
            break;
        end % if
    end % while
```

For additional information

- The MathWorks website:
  - http://www.mathworks.com