Matlab implementation notes

August 28, 2018

Master node of Risk_Sensitive_Reachability_Project Repository

---

Files

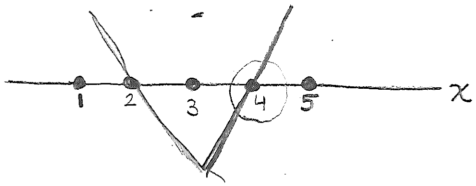
- Main.m - Main script
- CVaR_Bellman-Backup.m - does DP recursion
- maxExp.m - computes $\max_{\substack{R \in [0,\frac{1}{y}] \\ \text{etc.}}} \mathbb{E}\left[R \cdot J_{k+1}(x_{k+1}, yR) \,\middle|\, x_k, u_k, y\right]$ approximately
- getLMIs.m - gets linear matrix inequalities for maxExp.m
- getPossControls.m - returns control actions given current state
- stage-cost.m - $c(x) = e^{g(x)}$, $g(x)$ encodes constraint set

$dx = 1$

$x + u + w$

$5 - 1 + 1 = 5$

$4 - 1 + 1 = 4$



$g(x) = |x-3| - 1$

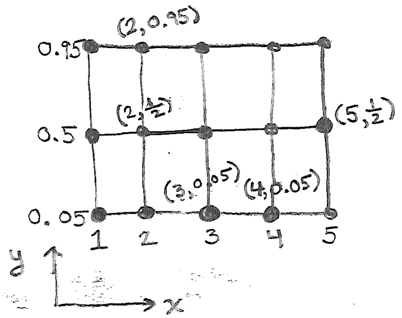
$$J_{N+1} = \begin{bmatrix} \overset{J_{N+1}(x_1,y_1)}{2.718} & \overset{J_{N+1}(x_2,y_1)}{1} & \overset{J_{N+1}(x_3,y_1)}{0.368} & 1 & 2.718 \\ \overset{J_{N+1}(x_1,y_2)}{2.718} & 1 & \overset{J_{N+1}(x_3,y_2)}{0.368} & \overset{J_{N+1}(x_4,y_2)}{1} & \overset{J_{N+1}(x_5,y_2)}{2.718} \\ \overset{J_{N+1}(x_1,y_3)}{2.718} & 1 & \overset{J_{N+1}(x_3,y_3)}{0.368} & 1 & 2.718 \end{bmatrix}$$

J_3 ✓

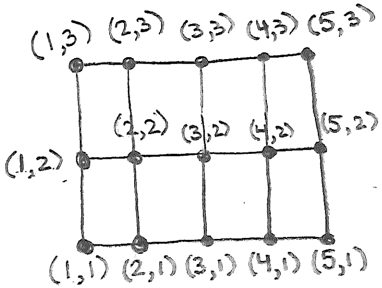
$e^{g(1)}$ $e^{g(2)}$ $e^{g(3)}$ $e^{g(4)}$ $e^{g(5)}$

smush together to get $\approx J(x_{next}, y_1)$

The values of each grid point



Coordinates of the grid.



	x_1	x_2	...	x_5	
$Js\{N+1\} =$	$J_N(x_1, l_1)$	$J_N(x_2, l_1)$	...	$J_N(x_5, l_1)$	$l_1 = 0.95$
	$J_N(x_1, l_2)$	$J_N(x_2, l_2)$	...	$J_N(x_5, l_2)$	$l_2 = ½$
	$J_N(x_1, l_3)$	$J_N(x_2, l_3)$	...	$J_N(x_5, l_3)$	$l_3 = 0.05$

$=$	$e^{g(x_1)}$	$e^{g(x_2)}$	...	$e^{g(x_5)}$
	$e^{g(x_1)}$	$e^{g(x_2)}$	...	$e^{g(x_5)}$
	$e^{g(x_1)}$	$e^{g(x_2)}$	...	$e^{g(x_5)}$

$e^{g(x_1)} = e^{g(1)} = e^1 = \boxed{2.718}$

$e^{g(x_2)} = e^{g(2)} = e^0 = \boxed{1}$

$e^{g(x_3)} = e^{g(3)} = e^{-1} = \boxed{0.368}$

$e^{g(x_4)} = e^{g(4)} = e^0 = \boxed{1}$

$e^{g(x_5)} = e^{g(5)} = e^1 = \boxed{2.718}$

$$[X, L] = \text{meshgrid}\Big(\underbrace{[1,2,3,4,5]}_{x_1\ x_2\ \cdots\ x_5}^{xs}, \underbrace{[0.95, \tfrac{1}{2}, 0.05]}_{l_1\ \ l_2\ \ l_3}^{ls}\Big)$$

$$X = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \end{array}$$

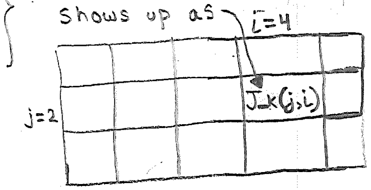
$j=2, i=4$

$X(j,i) = x$ (row col)

$L(j,i) = y$

Used to compute $J_k(x,y)$

Shows up as $i=4$



$= J\_k$ in CVaR_BellmanBackup.m

$$L = \begin{bmatrix} 0.95 & 0.95 & 0.95 & 0.95 & 0.95 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix} \begin{matrix} l_1 \\ l_2 \\ l_3 \end{matrix}$$

satisfies assumption 1 in Chow 2015

Our initial condition is:

$J_N(x,y) := e^{g(x)}$ is constant in y, so is also continuous in y,

$y J_N(x,y) = y \cdot e^{g(x)}$ is linear in y, so is also concave in y,

for any fixed x.

We computed: $\hat{J}_0(x,y) = \min_{\pi} \text{CVaR}_y\left[\sum_{k=0}^{2} e^{g(x_k)} \,\middle|\, x_0 = x, \pi\right]$ via dynamic programming back-up.

Next, we want $\underline{\text{ground truth}}$ (analytic or enumeration?): $\tilde{J}_0(x,y) = \min_{\pi} \text{CVaR}_y\left[\max_{k=0,1,2} g(x_k) \,\middle|\, x_0 = x, \pi\right]$

Then, compare how close. Should have $\hat{J}_0 \geq \tilde{J}_0$

Implementation notes: MaxExp.m

$\max\mathrm{Exp}(J_{k+1},\ x,\ u,\ y,\ xs,\ ls)$

$$xs = \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ [1,\ 2,\ 3,\ 4,\ 5] \end{array} \quad \text{discretized state space}$$

$$ls = \begin{array}{c} [0.95,\ \tfrac{1}{2},\ 0.05] \\ \begin{array}{ccc} l_1 & l_2 & l_3 \end{array} \end{array} \quad \text{discretized confidence levels}$$

x_1	x_2	...	x_5	
$J_{k+1}(x_1,l_1)$	$J_{k+1}(x_2,l_1)$	...	$J_{k+1}(x_5,l_1)$	$l_1 = 0.95$
$J_{k+1}(x_1,l_2)$	$J_{k+1}(x_2,l_2)$	...		$l_2 = 1/2$
$J_{k+1}(x_1,l_3)$	$J_{k+1}(x_2,l_3)$	...		$l_3 = 0.05$

Want to solve $\max\limits_{R} E\left[R \cdot J_{k+1}(x_{k+1}, yR) \,\middle|\, x_k, y, u_k\right]$ subject to $x_{k+1} = f(x_k, u_k, w_k)$

$P(w^j) = p_j$ is given for $j=1,\dots,m$

$R(w^j) \in [0, \frac{1}{y}]$, $\sum_{j=1}^{m} R(w^j)\, p_j = 1$

$$\|$$

$$\sum_{j=1}^{m} \underbrace{R(w^j)}_{r_j} \cdot J_{k+1}\Big(\overbrace{f(x_k,u_k,w^j)}^{x_{k+1}^j},\ yR(w^j)\Big) \cdot p_j$$

$$\|$$

$$\max_{r_1,\dots,r_m} \sum_{j=1}^{m} r_j \cdot J_{k+1}(x_{k+1}^j,\ yr_j) \cdot p_j$$

subject to $x_{k+1}^j = f(x_k,u_k,w^j)$

$r_j \in [0, \frac{1}{y}]$, $\sum_{j=1}^{m} r_j p_j = 1$

Change of variable

$r_j := z_j / y$

$z_j = y \cdot r_j$

$z_i = y r_i \in [0, 1]$

$$\sum_{j=1}^{m} \frac{z_j}{y} p_j = 1$$

$$\frac{1}{y} \sum_{j=1}^{m} z_j p_j = 1$$

$$\sum_{j=1}^{m} z_j p_j = y$$

$$\|$$

$$\max_{z_1,\dots,z_m} \sum_{j=1}^{m} \frac{z_j}{y} \cdot J_{k+1}\left(x_{k+1}^{j}, z_j\right) \cdot p_j \qquad \text{subject to } x_{k+1}^{j} = f(x_k, u_k, w^j)$$

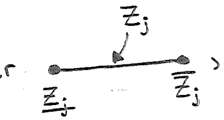
$$z_j \in [0,1]$$

$$\|$$

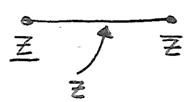
$$\sum_{j=1}^{m} z_j p_j = y$$

$$\max_{z_1,\dots,z_m} \frac{1}{y} \sum_{j=1}^{m} \boxed{z_j \cdot J_{k+1}\left(x_{k+1}^{j}, z_j\right)} \cdot p_j$$

$$\approx$$

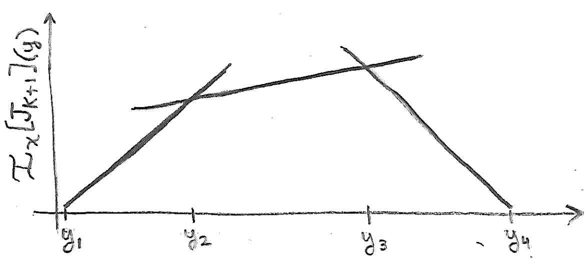
$\mathcal{I}_{x_{k+1}^{j}}\left[J_{k+1}\right](z_j)$ ← the linear interpolation of $z\, J_{k+1}(x, z)$ over 
where $x = x_{k+1}^{j}$ is fixed

Definition $\mathcal{I}_x\left[J_{k+1}\right](z) := \text{slope}_z \cdot z + \underline{z}\left(J_{k+1}(x, \underline{z}) - \text{slope}_z\right)$ (see Chow 2015, middle of page 6)



$$\text{slope}_z = \frac{\bar{z}\, J_{k+1}(x, \bar{z}) - \underline{z}\, J_{k+1}(x, \underline{z})}{\bar{z} - \underline{z}}$$

Fact



$\mathcal{I}_x\left[J_{k+1}\right](y)$ is continuous, piecewise linear, concave in $y \in (0, 1]$.
(see page 15 - Chow 2015 supplemental)

Fact (www.seas.ucla.edu/~vandenbe/ee236a/lectures/pwl.pdf) "Minimimizing a sum of piecewise linear functions slide"

My translation

maximize $f(x) + g(x) = \min_{i=1,\dots,m} (a_i^T x + b_i) + \min_{i=1,\dots,p} (c_i^T x + d_i)$

$$\Updownarrow$$

maximize $t_1 + t_2$ subject to $\underset{i=1,\dots,m}{a_i^T x + b_i \geq t_1}$, $\underset{i=1,\dots,p}{c_i^T x + d_i \geq t_2}$

What's written in the notes

minimize $f(x) + g(x) = \max_{i=1,\dots,m} (a_i^T x + b_i) + \max_{i=1,\dots,p} (c_i^T x + d_i)$

$\Updownarrow$ for fixed x, optimal t_1, t_2 are $t_1 = f(x)$, $t_2 = g(x)$.

minimize $t_1 + t_2$ subject to $\underset{i=1,\dots,m}{a_i^T x + b_i \leq t_1}$, $\underset{i=1,\dots,p}{c_i^T x + d_i \leq t_2}$

We will solve the approximate problem,

$$\max_{z_1,\ldots,z_m} \frac{1}{y} \sum_{j=1}^{m} \underbrace{\mathcal{I}_{x^j_{k+1}}\left[J_{k+1}\right](z_j)}_{\text{piecewise linear, concave in } z_j} \cdot p_j \qquad \text{subject to } x^j_{k+1} = f(x_k, u_k, w^j)$$

$$z_j \in [0,1]$$

$$\sum_{j=1}^{m} z_j p_j = y$$

weighted sum of piecewise linear, concave functions

$$\Updownarrow$$

corresponds to $As\{i\}Z(1) + bs\{i\} \geq t(1)$ in code.

$$\max_{\substack{z_1,\ldots,z_m \\ t_1,\ldots,t_m}} \frac{1}{y} \sum_{j=1}^{m} t_j \cdot p_j \qquad \text{subject to} \quad \text{LMIs for } \text{“}\mathcal{I}_{x^1_{k+1}}\left[J_{k+1}\right](z_1) \geq t_1\text{”}$$

$$\vdots$$

$$\text{“}\mathcal{I}_{x^m_{k+1}}\left[J_{k+1}\right](z_m) \geq t_m\text{”}$$

How do we express this exactly?

$$x^j_{k+1} = f(x_k, u_k, w^j) \quad \text{for } j = 1, \ldots, m$$

each disturbance sample

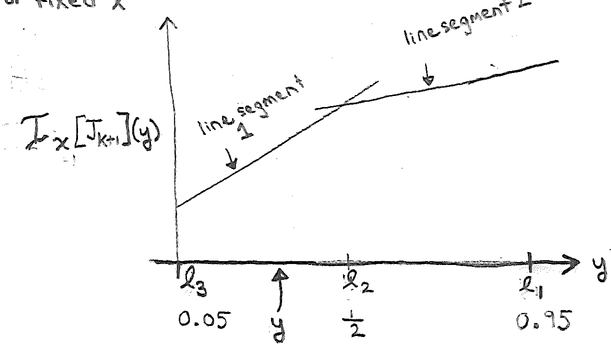
$$z_j \in [0,1], \quad \sum_{j=1}^{m} z_j p_j = y$$

m ↑ # of disturbance realizations (equiv. next state realizations)

in the code now $m=3$

getLMIs.m

For fixed x



$$\mathcal{I}_x[J_{k+1}](y) = \min_{j=1,2}\left\{ a_j y + b_j \right\} \geq t_1$$

$$\Updownarrow$$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} y + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \geq \begin{bmatrix} t_1 \\ t_1 \end{bmatrix}$$

t_1

each line segment exceeds t_1

$$\mathcal{I}_x[J_{k+1}](y) = l_{j+1} J_{k+1}(x, l_{j+1}) + \underbrace{\left(\frac{l_j J_{k+1}(x,l_j) - l_{j+1} J_{k+1}(x, l_{j+1})}{l_j - l_{j+1}}\right)}_{a_j(x)}(y - l_{j+1})$$

for $y \in [l_{j+1}, l_j]$

$$a_j(x) = \frac{l_j \cdot J_{k+1}(x, l_j) - l_{j+1} \cdot J_{k+1}(x, l_{j+1})}{l_j - l_{j+1}}$$

$$b_j(x) = l_{j+1}\left(J_{k+1}(x, l_{j+1}) - a_j(x)\right)$$

getLHS.m

$$x_{k+1}^{i} = x_k + u_k + ws(i)$$

We can't evaluate $J_{k+1}(x_{k+1}^{i}, l_j)$ or $J_{k+1}(x_{k+1}^{i}, l_{j+1})$ exactly because x_{k+1}^{i} may not be in the discretized state space.
($x_{k+1}^{i} \notin xs$, in general)

Instead, we interpolate

$$J_{k+1}(x_{k+1}^{i}, l_j) \approx \text{interp1}\Big(\underbrace{[x_1, x_2, x_3, x_4, x_5]}_{\text{"xs"}}, \underbrace{[J_{k+1}(x_1,l_j), J_{k+1}(x_2,l_j), J_{k+1}(x_3,l_j), J_{k+1}(x_4,l_j), J_{k+1}(x_5,l_j)]}_{\text{"J\_kPLUS1(j,:)" row j, all cols}}, x_{k+1}^{i}, \text{"linear"}\Big).$$

for $j = nl-1 : -1 : 1$

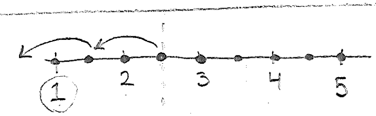
$j = nl-1$
- J^{j+1} = interp... J_kPLUS1(j+1, :) $j+1 = nl = 3$ $J_{k+1}(\cdot, l_3)$ ... $l_3 = 0.05$
- J^{j} = interp... J_kPLUS1(j, :) $j = nl-1 = 2$ $J_{k+1}(\cdot, l_2)$ ... $l_2 = 1/2$

$j = nl-2$
- J^{j+1} = ... $j+1 = nl-1 = 2$ $J_{k+1}(\cdot, l_2)$ $l_2 = 1/2$
- J^{j} = .. $j = nl-2 = 1$ $J_{k+1}(\cdot, l_1)$ $l_1 = 0.95$

getPossControls.m

$$x_{k+1} = x_k + u_k + w_k$$

$$0.5 = 2.5 - 1 - 1 \quad \text{not in grid!}$$



How to choose u, so we stop stepping out of grid

If we start at $x=3$, and $u_k = w_k = -1$. We take the biggest step as possible towards the boundary. Then,
$x_{k+1} = 3-1-1 = 1$, and we stay in grid.
But, if $x<3$ and $u_k = w_k = -1$, then we will step outside of grid.
So, we enforce $u_k = +1$ if $x<3$, so sys does not leave grid.

if $x < 3$ then $u_1 = 1$

if $x < \min + 2$, then $u = +1$

biggest step to the left (towards the boundary) in 1 time step.

if $x > \max - \boxed{2}$, then $u = -1$.

biggest step to the right (towards boundary) in 1 time step

$$2 = \underbrace{1}_{u_k} + \underbrace{1}_{w_k}$$