Assignment 4

Per Emil Hammarlund, Albert Öst

2019-05-10

Contents

Implementation of backward pass

The backward pass was implemented using the following code:

```
function betaHat=backward(mc,pX,c)
%--------------------------------------------------------
%Code Authors:
% Albert Öst
% Per Emil Hammarlund
%--------------------------------------------------------

%Initialization step
T=size(pX,2);%Number of observations

betaHat = zeros(size(pX));

% Depending on if the HMM is finite or not, we get different computations
isFinite = finiteDuration(mc);
if isFinite
    betaHat(:, end) = mc.TransitionProb(:, end)./(c(end-1)*c(end));
    A = mc.TransitionProb(:, 1:end - 1);
else
    betaHat(:, end) = ones(size(pX, 1), 1)./c(end);
    A = mc.TransitionProb;
end


% Backward step
for t = T-1:-1:1
    betaHat(:, t) = A * (pX(:, t + 1) .* betaHat(:, t + 1))  ./ c(t);
end

end
```

Validation of backward pass

The validation of the backward pass was implemented using the following code:

```
format long
clear

% Observations
x = [-0.2 2.6 1.3];

% Infinite
%mc = MarkovChain([0.75; 0.25], [0.99 0.01; 0.03 0.97]);
% Finite
mc = MarkovChain([1; 0], [0.9 0.1 0; 0 0.9 0.1]);
g1 = GaussD('Mean', 0, 'StDev', 1);
g2 = GaussD('Mean', 3, 'StDev', 2);

pX = prob([g1, g2], x);

[~, c] = mc.forward(pX);

betaHat = mc.backward(pX, c)
```

Result from the test run

The test run gave the following result:

```
betaHat =

   1.000000000000000   1.038935709330079                   0
   8.415379245573641   9.350421383970712   2.081827732555444
```

Which was very close to the desired values in the lab instruction. Here c was calculated using the value returned from the forward algorithm. In the lab instruction c was rounded to 4 decimal points in each element. c had the following values from the forward pass:

```
c =

   1.000000000000000
   0.162523466100529
   0.826580955035720
   0.058112534334093
```

These values where rounded to 4 decimal points, and c was assignted them:

```
c = [1.0000, 0.1625, 0.8266, 0.0581];
```

And the same code was used again, this gave the following result:

```
betaHat =

   1.000337126754333   1.039285962353727                   0
   8.418216295065184   9.353573661183537   2.082228884429218
```

Which are the exactly the values that where desired in the lab instruction!