CPSC 532W Homework 5

Naomi Graham

March 18, 2021

All the code can be found on: `https://github.com/n6graham/cpsc532_hw5`.

1 Program 1

I printed the terminal output from running the unit tests to give incontrovertible proof that they all passed.

```
foppl deterministic test 1
foppl deterministic test 2
foppl deterministic test 3
foppl deterministic test 4
foppl deterministic test 5
foppl deterministic test 6
foppl deterministic test 7
foppl deterministic test 8
foppl deterministic test 9
foppl deterministic test 10
foppl deterministic test 11
foppl deterministic test 12
foppl deterministic test 13

FOPPL Tests passed
```

```
hoppl deterministic test 1

Test passed
hoppl deterministic test 2

Test passed
hoppl deterministic test 3

Test passed
hoppl deterministic test 4

Test passed
hoppl deterministic test 5

Test passed
hoppl deterministic test 6

Test passed
hoppl deterministic test 7

Test passed
hoppl deterministic test 8

Test passed
hoppl deterministic test 9

Test passed
hoppl deterministic test 10

Test passed
hoppl deterministic test 11

Test passed
hoppl deterministic test 12

Test passed
All deterministic tests passed
```

```
probabilistic test 1

('normal', 5, 1.4142136)
p value 0.022513078512681146
probabilistic test 2

('beta', 2.0, 5.0)
p value 0.40060626577281777
probabilistic test 3

('exponential', 0.0, 5.0)
p value 0.5945341474290768
probabilistic test 4

('normal', 5.3, 3.2)
p value 0.6784377129193204
probabilistic test 5

('normalmix', 0.1, -1, 0.3, 0.9, 1, 0.3)
p value 0.8861207327546117
probabilistic test 6

('normal', 0, 1.44)
p value 0.5685375949870255
All probabilistic tests passed
```

5 Snippets of code

```
class Env(dict):
    "An environment: a dict of {'var': val} pairs, with an outer Env."
    def __init__(self, parms=(), args=(), outer=None):
        self.update(zip(parms, args))
        self.outer = outer
        #if outer is not None:
        #    self.outer = copy.deepcopy(outer)
    def find(self, var):
        "Find the innermost Env where var appears."
        return self if (var in self) else self.outer.find(var)
    def check_in_env(self, var):
        return (var in self) or (var in self.outer)

```

```
class Lambda(object):
    "A user-defined Scheme procedure."
    def __init__(self, parms, body, env):
        self.parms, self.body, self.env = parms, body, env
    def __call__(self, *args):
        return eval(self.body, Env(self.parms, args, self.env))

```

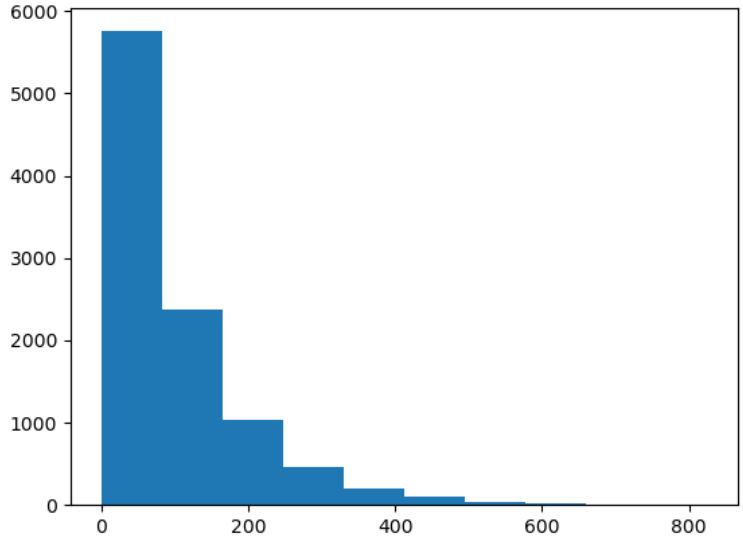
```
def standard_env() -> Env:
    env = Env()
    env.update(pmap(penv))
    return env

```

```
def eval(expr, envr):

        if type(expr) is torch.Tensor:
            return expr


        if is_const(expr, envr):
            if type(expr) in [int, float, bool]:
                expr = torch.Tensor([expr]).squeeze()
            elif type(expr) is torch.Tensor:
                return expr
            return expr


        if type(expr) is str: # and expr != 'fn':    # variable reference
            try:
                f = envr.find(expr)[expr]
                return f
            except AttributeError:
                return expr


        op, *args = expr

        if is_fn(op,envr):
            (params, body) = args
            local_env = Env(outer=envr)
            lam = Lambda(params,body,local_env)
```

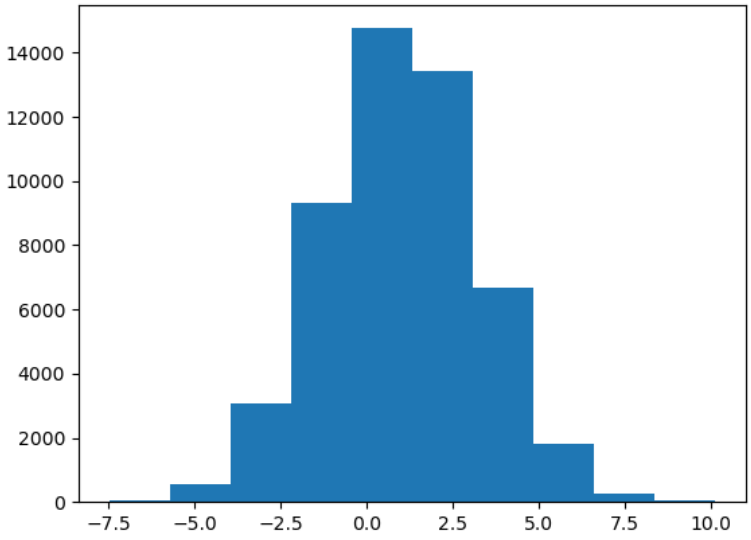
```
                return lam


            elif is_if(expr,envr):
                cond_expr, true_expr, false_expr = args[0], args[1], args[2]
                tf = eval(cond_expr,envr)
                res = (true_expr if tf else false_expr)
                return eval(res,envr)



            elif is_sample(expr,envr):
                dist_expr = args[1]
                dist_obj = eval(dist_expr,envr)
                s = dist_obj.sample()
                return s


            elif is_observe(expr,envr):
                #dist_expr, obs_expr = args[1], args[2]
                #dist_obj = eval(dist_expr,envr)
                #obs_value = eval(obs_expr,envr)

                return eval(args[-1],envr)


            else:
                proc=eval(op,envr)
                vals = [ eval(arg,envr) for arg in args]
                return proc(*vals)

```

```
    def evaluate(ast, envr=None):
        if envr is None:
            envr = standard_env()

        return eval([ast,'0'],envr)

```

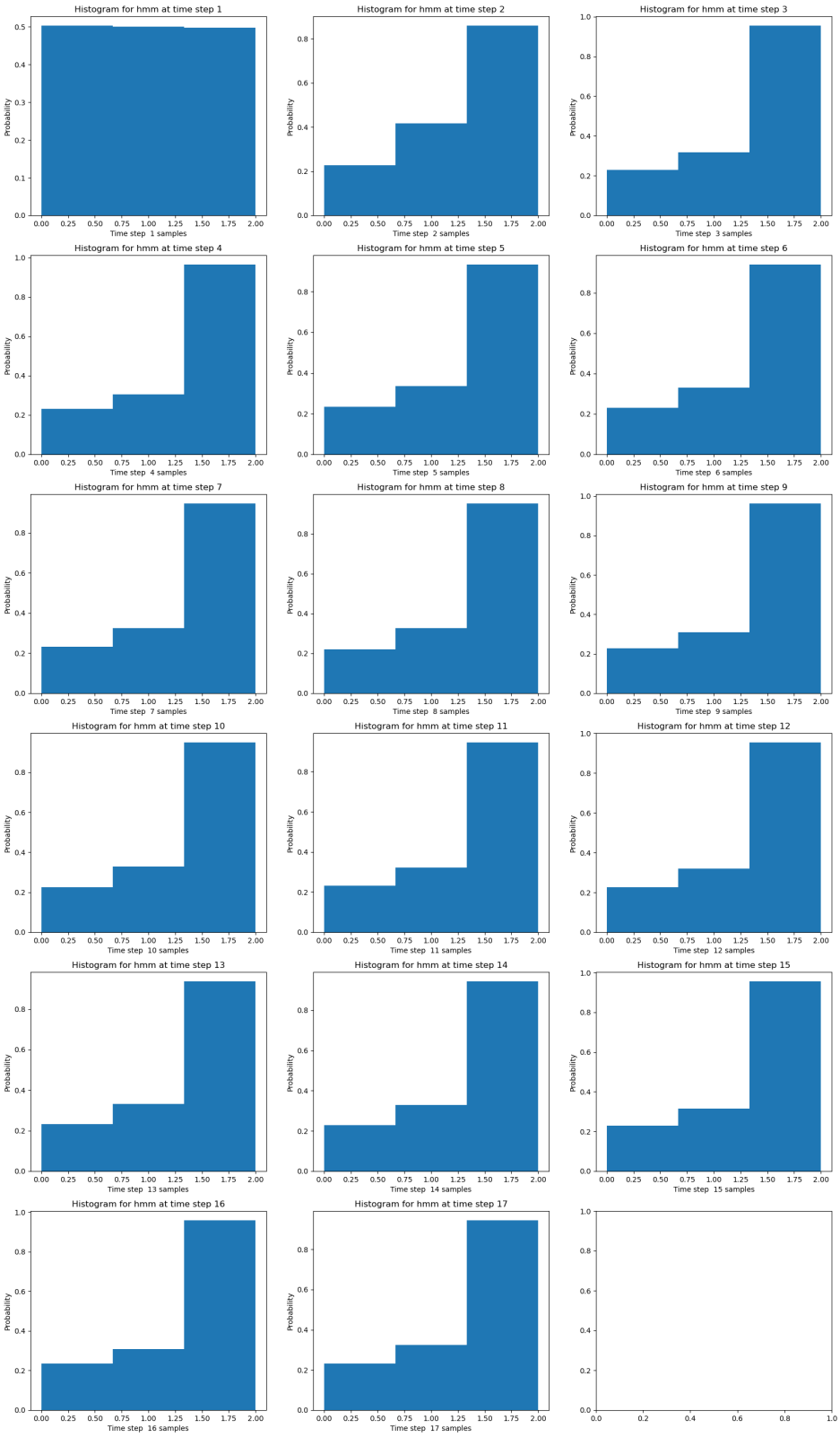
2 Program 2



3 Program 3



4 Program 4


Histogram for hmm at time step 1
Time step 1 samples
Histogram for hmm at time step 2
Time step 2 samples
Histogram for hmm at time step 3
Time step 3 samples
Histogram for hmm at time step 4
Time step 4 samples
Histogram for hmm at time step 5
Time step 5 samples
Histogram for hmm at time step 6
Time step 6 samples
Histogram for hmm at time step 7
Time step 7 samples
Histogram for hmm at time step 8
Time step 8 samples
Histogram for hmm at time step 9
Time step 9 samples
Histogram for hmm at time step 10
Time step 10 samples
Histogram for hmm at time step 11
Time step 11 samples
Histogram for hmm at time step 12
Time step 12 samples
Histogram for hmm at time step 13
Time step 13 samples
Histogram for hmm at time step 14
Time step 14 samples
Histogram for hmm at time step 15
Time step 15 samples
Histogram for hmm at time step 16
Time step 16 samples
Histogram for hmm at time step 17
Time step 17 samples
Probability