EE365 Homework 7

1. *The squared norm of a linear function.* Download an updated version of the class file `linear_function.m` from the course website.

   (a) Let $f(x) = Ax + b$ be a linear function. Show that $g(x) = \|f(x)\|^2$ is a quadratic function, and express the coefficients of g in terms of the coefficients of f.

   (b) Implement the function `norm_squared_linear` in the class `linear_function`. The file `linear_quadratic_extensions_data.m` contains a matrix A, a vector b, and a vector x. Evaluate $\|f(x)\|^2$ in two ways: first, evaluate f at x, and then compute the squared norm of the resulting vector; second, use the function `normsq` to compute g, and then evaluate g at x. Report the values of $\|f(x)\|^2$ that you find.

2. *Partial evaluation and expectation of a quadratic function.* Download an updated version of `quadratic_function.m` from the course website. Be sure to fill in all of the functions that you implemented in homework 6.

   (a) Consider a quadratic function

   $$f(x, y) = \tfrac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} q_x \\ q_y \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} + \tfrac{1}{2} r.$$

   Let $g_y(x)$ denote the value of $f(x, y)$ as a function of x for a fixed value of y: that is,

   $$g_y(x) = f(x, y).$$

   Show that $g_y(x)$ is a quadratic function of x, and find expressions for the coefficients of $g_y(x)$ in terms of the coefficients of $f(x, y)$.

   (b) Implement the function `partial_evaluation` in the class `quadratic_function`. The file `linear_quadratic_extensions_data.m` contains a matrix `P`, vectors `x`, `y`, and `q`, and a scalar `r`. Evaluate $f(x, y)$ in two ways: first, compute g_y, and then evaluate g_y at x; second, evaluate f at (x, y). Report the values you find using these two methods.

   Note. The `quadratic_function` class is configured so that `qf(y,m)` computes the partial evaluation of the quadratic function `qf` with the last `m` entries of the argument set equal to `y`.

   (c) Recall that the expectation of a random matrix is defined entrywise, and the variance of a random vector x is defined to be

   $$\mathbf{var}(x) = \mathbf{E}((x - \mathbf{E}(x))(x - \mathbf{E}(x))^T).$$

   For a random vector x, and a constant matrix A, show that

   $$\mathbf{E}(x^T A x) = \mathbf{E}(x)^T A\, \mathbf{E}(x) + \mathbf{Tr}(A\, \mathbf{var}(x)).$$

Hint. A scalar is equal to its own trace, and the trace satisfies $\mathbf{Tr}(AB) = \mathbf{Tr}(BA)$ whenever the products AB and BA are both defined.

(d) Show that the expectation of $f(x, y)$ over y is

$$h(x) = \mathop{\mathbf{E}}_{y}(f(x, y)) = g_{\mathbf{E}(y)}(x) + \tfrac{1}{2}\,\mathbf{Tr}(P_{yy}\,\mathbf{var}(y)).$$

(e) Implement the function `partial_expectation` in the class `quadratic_function`. The file `linear_quadratic_extensions_data.m` contains a matrix `P`, vectors `x` and `q`, a scalar `r`, a cell array `yvals`, and a vector `ypmf`. Each entry of `yvals` is a value for the vector y, and the corresponding component of `ypmf` gives the probability mass of that value. Evaluate $\mathbf{E}_y(f(x, y))$ in two ways: first, compute h, and the evaluate h at x; second, compute the expectation explicitly by computing the weighted sum of $f(x, y)$ over the values of y. Report the values you find using these two methods.

Note. The `quadratic_function` class is configured so that `Expect(qf,ybar,yvar)` computes the partial expectation of the quadratic function `qf` with respect to the last `length(ybar)` entries of the argument, when those entries have mean `ybar`, and variance `yvar`.

3. *A special case of linear/quadratic stochastic control.* Consider a linear dynamical system with process noise:

$$x_{t+1} = f(x_t, u_t, w_t) = Ax_t + Bu_t + w_t.$$

We assume that $x_0, w_0, w_1, w_2, \ldots$ are independent, and that the mean and variance of w_t are time-invariant and known. Suppose the stage cost is a convex quadratic function of x and u:

$$g(x, u) = \frac{1}{2}\begin{bmatrix} x \\ u \\ 1 \end{bmatrix}^T \begin{bmatrix} P_{xx} & P_{xu} & q_x \\ P_{ux} & P_{uu} & q_u \\ q_x^T & q_u^T & r \end{bmatrix} \begin{bmatrix} x \\ u \\ 1 \end{bmatrix}.$$

We want to find a steady-state controller and value function for this system.

(a) Since this is a linear/quadratic stochastic control problem, we know that the optimal controller is linear, and the optimal value function is quadratic. Suppose $v_0(x)$ is any quadratic function of x. We can find the optimal controller and value function using value iteration:

$$v_{k+1}(x) = \mathcal{T}(v_k) = \min_{u}\{g(x, u) + \mathop{\mathbf{E}}_{w}(v_k(f(x, u, w)))\}, \qquad k = 0, 1, 2, \ldots.$$

We repeat value iteration until convergence: that is, until the controller and value function do not change much from one iteration to the next. Write a `MATLAB` function to compute the optimal steady-state controller and value function; use the following function header.

```
function [val , pol] = ss_lqsc(f,g,wbar,wvar)
```

Here, `f` is a `linear_function` specifying the dynamics as a linear function of (x, u, w), `g` is a `quadratic_function` specifying the stage cost function as a quadratic function of (x, u), and `wbar` and `wvar` are the mean and variance of w_t, respectively. We are *not* asking you to work through the algebra: let your `linear_function` and `quadratic_function` classes do most of the work.

(b) Apply your `ss_lqsc` function to the instance of the problem defined in the file `ss_lqsc_data.m`. Report the coefficients of the optimal steady-state controller, and the quadratic and linear coefficients of the optimal steady-state value function.

4. *Managing a crossbar-switch router.* Consider a router with m input ports, and n output ports. Packets arrive at the input ports, and must be sent to the output ports. In particular, each packet arrives at a specific input port, and must be sent to a specific output port. In each time period, the number of packets that arrive at input port i, and must be sent to output port j is a Poisson random variable with rate parameter λ_{ij}. We assume that these random variables are all independent.

As packets arrive, they are stored in buffers at the input ports. There is a separate buffer for each input port; packets are sorted by destination in these buffers, and each buffer can hold at most B packets with each destination. (In other words, each input buffer can hold B packets that must be sent to output port 1, B packets that must be sent to output port 2, and so on. Thus, the total capacity of the buffer is nB. However, the buffer cannot store an arbitrary collection of nB packets since it can store at most B packets with each destination.) Packets that overflow a buffer are dropped.

After the packets have arrived, and packets that overflow a buffer have been dropped, the router must decide what to do in the current time period. We assume that the router is a crossbar switch, which means that, in each time period, we can connect each input port to at most one output port, and we can connect at most one input port to each output port. If the buffer of the input port contains at least one packet whose destination is the output port to which the input port is connected, then one packet is sent from the input port to the output port; otherwise, nothing happens.

We pay a penalty $C_b > 0$ for each packet in an input buffer at the end of a time period, and we pay a penalty $C_d > 0$ for each dropped packet. We receive a reward $C_r > 0$ for each packet that is successfully routed to its destination. All the parameters for this problem are defined in `crossbar_switch_data.m`. Additionally, the file `hw7_coding_tips.m` contains some implementation remarks that you may find useful.

(a) Formulate the problem of managing the crossbar-switch router as a Markov decision problem. Is dynamic programming tractable for this problem?

(b) Consider the following heuristic policy. In each time period, we find the input/output pair with the largest number of packets currently in the buffer (we break ties arbitrarily). We link this input/output pair. Ignoring input and output ports that have already been linked, we find the input/output pair with the largest number of packets currently in the buffer, and link this input/output pair. We repeat this procedure until either all of the input ports or all of the outputs ports have been linked. Simulate this policy for T time steps, and report the average stage cost.

(c) Consider a linear/quadratic stochastic control problem with dynamics

$$f(x, u, w) = x - u + w,$$

and stage cost

$$g(x, u, w) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{ij} x_{ij}(t)^2 + \frac{\rho_1}{2mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(u_{ij}(t) - \frac{1}{2} \right)^2 + \frac{\rho_2}{2m} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} u_{ij}(t) - 1 \right)^2 + \frac{\rho_3}{2n} \sum_{j=1}^{n} \left(\sum_{i=1}^{m} u_{ij}(t) - 1 \right)^2.$$

where $\rho_1, \rho_2, \rho_3 \geq 0$ are parameters. Comment on this linear/quadratic stochastic control problem as an approximation of the problem of managing the crossbar-switch router. What aspects of the problem are modeled well? What approximations are made? What is the purpose of each term in the cost function?

(d) Use your function `ss_lqsc` to compute the optimal steady-state controller and value function for the linear/quadratic stochastic control (LQSC) problem in 4c. Propose a method for rounding the LQSC action to a feasible action for the actual problem. Simulate the policy obtained by rounding the LQSC action; try different values of the ρ parameters. Report the best average stage cost that you find, and the values of the ρ parameters that you used.

(e) Implement approximate dynamic programming for this problem. Use the optimal steady-state value function of the LQSC approximation as your approximate value function. Simulate the ADP policy with different values of the ρ parameters. Report the best average stage cost that you find, and the values of the ρ parameters that you used.