LV 11.4500 – UBUNG 3

Exercises from FJK. 2.2.9, 2.2.10, 2.2.19, 2.2.35, 2.2.36, 2.2.40, 2.3.9, 2.3.11,

Other exercises.

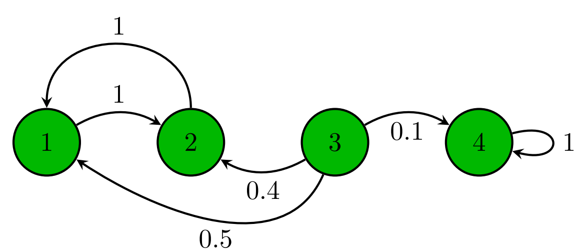
U3.1 For the transition function p depicted below, determine which states that are

a) aperiodic

b) recurrent

and

c) describe the asymptotic behavior of $\pi^n = \pi^{n-1}p$ when $\pi^0(j) = \mathbb{1}_{\{3\}}(j)$. Will it converge towards an invariant distribution?



U3.2 Consider the following Example 8 from Lecture 6: Let X_n be a simple symmetric RW on $\mathbb{Z}$ and $Y_n = X_n + W_n$, where $\{W_n\}$ is iid and independent of $\{X_n\}$ with $\mathbb{P}(W_n = k) = 1/5$ for all $|k| \leq 2$. Assume $X_0 = 0$. Compute the value of $\mathbb{P}(X_2 = 0 \mid Y_{0:2} = (0, 2, 1))$ using Algorithm 1 (preferably on a computer, as the alternative is tedious).

Hint: First verify that

$$q^{ra}(c, d) = \frac{\mathbb{1}_{|a-r|=1}\mathbb{1}_{|d-a|\leq 2}}{10}$$

U3.3 Programming exercise on filtering: Consider a Markov chain $\{(X_n, Y_n)\}_n$ similar to that in U3.2, but with the single exception that that now $\mathbb{P}(W_n = k) = 1/11$ for all $|k| \leq 5$. We are given the following sequence of observations $Y_{1:10}$:

```
y = [4,-3, 4, -4, -6,  3,  3, -5, -7, -1];
```

Since Matlab's index counter starts at 1 rather than 0, we align the convention that $n = 1$ denotes the beginning of time in this exercise, and thus with the random walk initial condition $X_1 = 0$. The vector y has been generated using the synthetic data x: first one realization of the simple symmetric random walk $X_{1:10}$ was generated, taking the values

```
x = [0, 1, 0, -1, -2, -1, -2, -3, -2, -3];
```

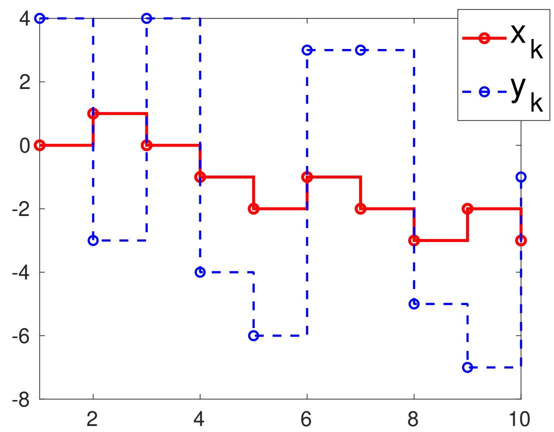


FIGURE 1. The synthetic data "unobserved process" (red line) and the observations of said process (dashed blue) for U3.3

and thereafter

$$Y_{1:10} = X_{1:10} + W_{1:10} \text{ or, if you like, } y_{1:10} = x_{1:10} + w_{1:10}.$$

The task of this exercise is to compute the vector $\tilde{x}_n = \mathbb{E}[X_n \mid Y_{0:n}]$ and to verify by eye measure that $\tilde{x}_n$ tends to approximate the underlying signal x_n better than y_n does. In Matlab, the jump-discontinuous paths x, y and $\tilde{x}$ are conveniently plotted (without under-the-hood linear interpolation) using the **stairs** function:

```
stairs(y,'--ob');hold on; stairs(x,'-or'); stairs(tildeX)
```

The output of the above commands, excluding the solution $\tilde{x}$ of this exercise, is presented in Figure 1.

As a second comparison, compute the following squared path errors over time:

$$\text{Error}(\tilde{x}) = \sum_{k=1}^{10} (\tilde{x}_k - x_k)^2$$

and

$$\text{Error}(y) = \sum_{k=1}^{10} (y_k - x_k)^2$$

$$\text{Hint: First verify that} \quad q^{ra}(c,d) = \frac{\mathbb{1}_{|a-r|=1}\mathbb{1}_{|d-a|\le 5}}{22}.$$

Remark: This kind of study is of course typically not a possible in practice, since you rarely have access to the "unobserved process" X. Nevertheless, usage of synthetic data is a good tool to test the performance of filtering methods.