DD2447 Statistical Methods in Applied Computer Science Project

Mateusz Buda, 921028-T177

January 26, 2017

1 Description

The project models railway with a single train. The base railway is a 3-regular graph. The railway graph is directed and each edge corresponds to one direction from set $\{L, R, 0\}$. Then, each node corresponds to switch that can be set either to the left or right. If the train enters a node from incoming edge 0, it goes in the direction of switch. If the incoming edge is either L or R, the train always goes in the 0 direction. When the train passes a node, we receive incoming or outgoing direction of the train and switch setting. Each observation is one from the following set $\{0L, 0R, L0, R0\}$. $0L$ and $0R$ observations mean the the train arrived from direction 0 and its switch is set to L or R respectively, whereas observations $L0$ and $R0$ mean that the switch is set to L or R and the train leaves in the direction 0. However, the signal of each switch setting (L or R) is noisy with probability $p = 0.05$. The 0 direction is always send correctly. Given the railway model with labeled edges and (noisy) observation sequence from each node, the task is to estimate the final train position.

2 Formal problem formulation

2.1 Notation

G	Railway directed graph
σ	Switches setting function $V(G) \to \{L, R\}$
O	Observations / emissions sequence
s	train stop position

2.2 Method

As given in provided project description the train stop position will be computed for each node $v \in V(G)$ as follows

$$P(s|G,O) = \sum_{\sigma} \frac{P(s,O|G,\sigma)P(\sigma|G,O)}{P(O|G,\sigma)}$$

and then with simple maximum likelihood we can pick one.

To compute $P(s,O|G,\sigma)$ I will use DP algorithm that with smart definition of HMM model is equivalent to HMM forward pass (α variable). From it I can also extract $P(O|G,\sigma)$ by summing probabilities for all final states. To estimate $P(\sigma|G,O)$ I will use suggested MCMC method, concretely, Metropolis–Hastings algorithm.

2.3 HMM definition

The whole solution will be based on the HMM definition. The number of hidden states is $2 \cdot |V(G)|$. For each node v there are two states. State $2v$ corresponds to being at node v and arriving via edge 0, whereas state $2v+1$ corresponds to being at node v and arriving via edge L or R. State 1 corresponds to train start position, does not emit any observation and its transition probability to other states is uniform for even states and 0 for others (the decision to have the initial probability vector represented as a separate state was made only to match MatLab HMM interface). This corresponds to the assumption that the train enters the railway from edge 0 and its first move is either L or R.

State transition probability matrix is binary, since for given σ we know the next state (position). There are four possible emissions $\{0L, 0R, L0, R0\}$. However, at odd states only $0L$ and $0R$ have non-zero probability, according to the switch setting. And for even states $L0$ and $R0$ have non-zero probabilities.

2.4 Observation and stop state probability computation with DP

With HMM definition as above, I can simply do a forward pass and extract all information that I need from α variable that is of size $(2 \cdot |V(G)|) \times T$, where T is observation sequence length. $\alpha(i,j)$ stores probability of being in state i and observing $\{o_1, \ldots, o_j\}$. Thus, the last column contains stop state probability distribution and the sum of last column is probability of a sequence being generated by given model. By merging neighbouring odd and even states, stop position probability distribution, given the emissions sequence can be obtained.

2.5 Switches setting distribution estimation

Following project description, to estimate $P(\sigma|G,O)$ I used Metropolis-Hastings MCMC method. The model σ with multivariate Bernoulli distribution, that is special case of binomial distribution, where the number of trials is 1. The

number of variables equals the number of nodes (switches) and corresponds to their setting. I assume that setting R is a success (in a Bernoulli distribution sense) and corresponds to variable being set to 1, whereas L corresponds to 0. Each variable is initialized with Gaussian distribution $\mathcal{N}(0.5, 0.1)$.

To estimate the likelihood required in M-H algorithm to compute acceptance probability, I sample the switches setting from given distribution, and compute observation likelihood from it. This is repeated a few times, and after discarding two extremes, the average is taken from the remaining values.

3 Experiments and analysis

I will consider the following 3 cases:

1. random graph (in terms of directions and switches setting) with 6 nodes;
2. graph with 6 nodes and all switches set to R that is "hamiltonian", i.e. regardless of starting position it will eventually start making circles around all nodes;
3. graph with 8 nodes, with half of the switches set to L and the other half to R, that is also "hamiltonian" and depending on the start position it will be making circle either in R or L direction.

For each of them, I generate 20 observations and estimate stop position probabilities. As an important intermediate result, I also compare true switches setting with the estimated one.

3.1 Random railway with 6 nodes

Figure 1 shows the random railway graph. The table below shows observation and node (and not state) for each time-steps from 1 to 20.

time-step	1	2	3	4	5	6	7	8	9	10
Observation	0L	R0	0L	R0	0L	R0	0L	R0	0L	R0
state	6	2	6	2	6	2	6	2	6	2

11	12	13	14	15	16	17	18	19	20
0L	R0	0L	R0	0L	R0	0L	R0	0L	R0
6	2	6	2	6	2	6	2	6	2

All of the observation are correct. However, the common scenario for random graph is that 2 edges between the same two nodes are L and 0 or R and 0. In that case, whenever the train enters one of them, it gets stuck between them. Therefore, it is not possible to correctly estimate switches setting function since the train does not explore the whole railway and only small part of it.

The correct switches setting for this graph comparing to the multivariate Bernoulli distribution parameters obtained from MCMC is

$$[0, 1, 0, 1, 1, 0] \leftrightarrow [0.7274, 0.8009, 1.0000, 0.9590, 0.3882, 0.1370]$$

They are very different from true distribution because there are other possible assignments that generate the same observation sequence.

The estimated stop position probability distribution is

$$[0.3028, 0.3031, 0.1724, 0.0182, 0.0000, 0.2036]$$

It gives the highest probability to nodes 1 and 2. Nevertheless, given that the switch setting estimation is totally incorrect, the stop position probabilities will not tell anything.

3.2 "Hamiltonian" graph with 6 nodes and all switches set R

Figure 2 shows the 6 nodes "hamiltonian" railway graph. The table below shows observation and node (and not state) for each time-steps from 1 to 20.

time-step	1	2	3	4	5	6	7	8	9	10
Observation	0R	R0	0R	R0	0R	R0	0R	R0	0R	L0
state	3	4	5	6	1	2	3	4	5	6

11	12	13	14	15	16	17	18	19	20
0R	R0	0R	R0	0R	R0	0R	R0	0R	R0
1	2	3	4	5	6	1	2	3	4

There is only one noisy observation at position 10. As the railway definition implies, from the very beginning the train goes in circle, starting from node 3.

The correct switches setting for this graph comparing to the multivariate Bernoulli distribution parameters obtained from MCMC is

$$[1, 1, 1, 1, 1, 1] \leftrightarrow [1.0000, 1.0000, 1.0000, 0.9020, 0.8863, 1.0000]$$

This time the estimated distribution parameters for switches setting is very accurate. However, the task was quite easy as all switches have the same setting. Hence, the states are indistinguishable. Every start and end position is equally probable.

The estimated stop position probability distribution is

$$[0.0000, 0.4535, 0.4458, 0.0535, 0.0000, 0.0472]$$

It gives the highest probability to nodes 2 and 3. In theory, it should be equal for all positions.

3.3 "Hamiltonian" graph with 8 nodes and half of the switches set to L and half to R

Figure 3 shows the 8 nodes railway graph. The table below shows observation and node (and not state) for each time-steps from 1 to 20.

time-step	1	2	3	4	5	6	7	8	9	10
Observation	0L	R0	0L	R0	0L	R0	0L	R0	0L	L0
state	3	4	1	2	7	8	5	6	3	4

11	12	13	14	15	16	17	18	19	20
0L	R0	0L	R0	0R	R0	0L	R0	0L	R0
1	2	7	8	5	6	3	4	1	2

There are two noisy observation at time-step 10 and 15. As the railway definition implies, from the very beginning the train goes in circle in L direction because the starting node 1 is set to L.

The correct switches setting for this graph comparing to the multivariate Bernoulli distribution parameters obtained from MCMC is

$$[0, 1, 0, 1, 0, 1, 0, 1] \leftrightarrow [0.0000, 0.9359, 0.0000, 1.0000, 0.1628, 1.0000, 0.0000, 0.8474]$$

This time the estimated distribution parameters for switches setting also is very accurate. And this time the task was not so easy bu still all even and all odd states are equal. It should be only possible able to tell if the train started in one from even or odd nodes.

The estimated stop position probability distribution is

$$[0.0024, 0.0474, 0.0000, 0.9046, 0.0000, 0.0000, 0.0000, 0.0456]$$

The train started in node 3 and stopped in 2. Almost entire probability mass for estimated stop position is in the even states that agrees with out expectation.

4 Remarks

As my project partner has dropped the course he did not contribute at all to any work presented here. Doing the project alone, I did not implement provided DP algorithm and instead of that I used special HMM definition together with functionality provided with MatLab to compute HMM forward pass α variable. Also to compute observation likelihood, MatLab HMM tool was used. For example graph generation I used another MatLab Tools for Network Analysis (http://strategic.mit.edu/downloads.php?page=matlab_networks). The rest of the code was written by me from scratch.

To make the results here full reproducible, I created them with fixed random number generator seed. The the code provided that ed with this report contains MatLab script `demo.m` allows to run all experiments presented in this report.

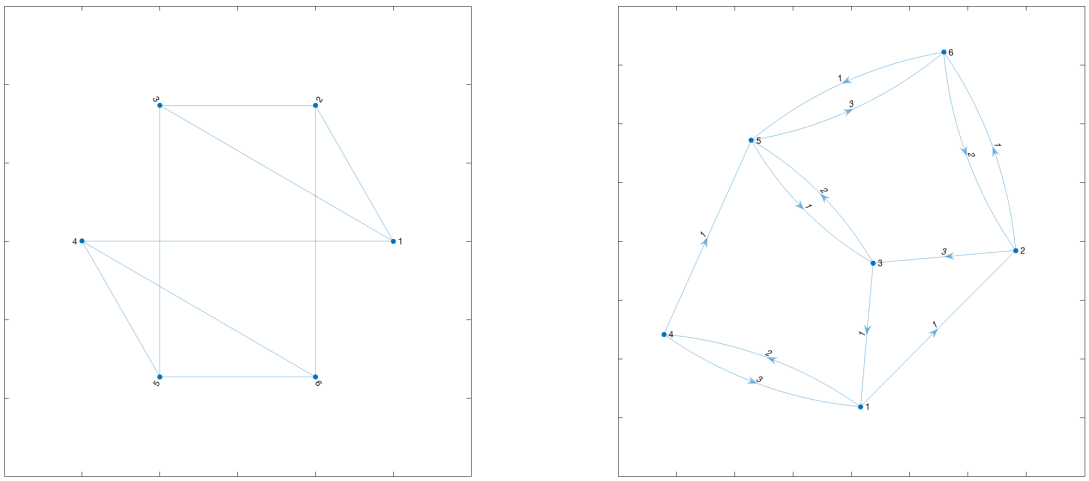

Figure 1: Random railway graph (in terms of directions and switches setting) with 6 nodes. Left graph is the base one and the right one show switches settings. Edge labels correspond to directions: 0 - 1, L - 2, R - 3. The same convention applies to all railway plots.

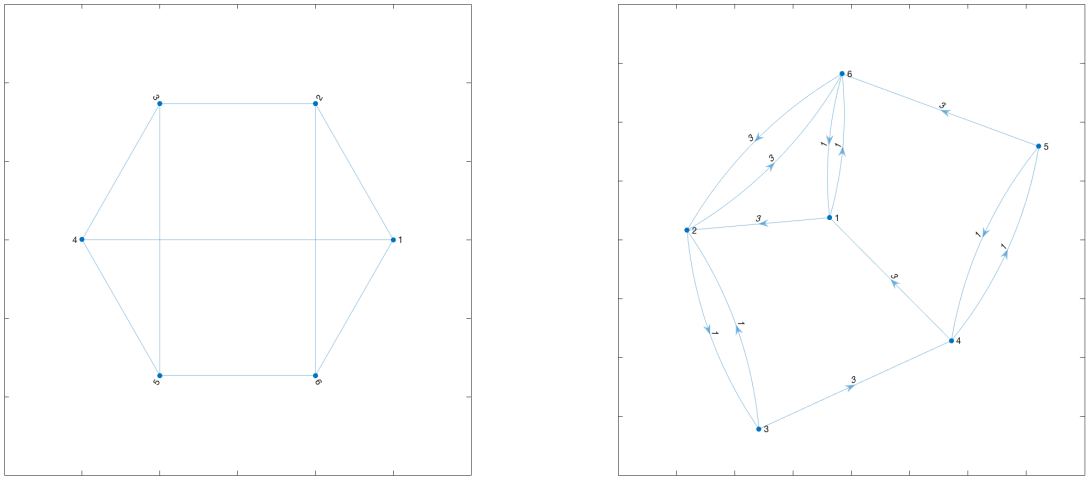

Figure 2: Graph with 6 nodes and all switches set to R that is "hamiltonian", i.e. regardless of starting position it will eventually start making circles around all nodes.

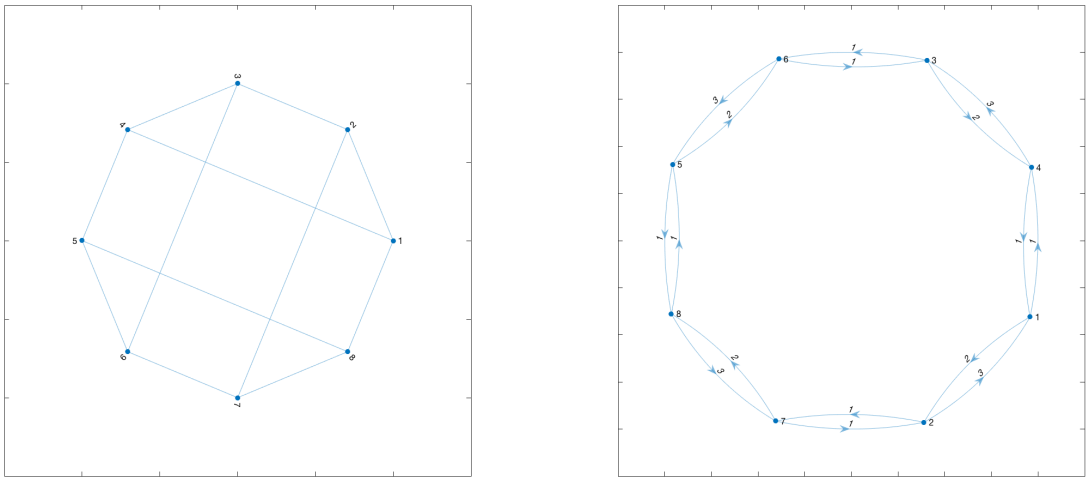

Figure 3: Graph with 8 nodes and half of the switches set to R and the other to L that is "hamiltonian". Depending on the starting node, the train makes circles either in L or R direction.