HMMs: Assignment 2

DD2434 Machine Learning, Advanced Course

Seger, Cedric
cedrics@kth.se

January 31, 2018

Contents

Dependencies in a Directed Graphical Model

Question 1

If we know X then nodes (A, B) become dependent due to the explaining away effect. Also node pairs (C, E) and (D, F) have dependencies due to their parent-child relationship.

Question 2

If X is not known, then it opens up a causal chain from top to bottom. We therefore have the following dependent pair of nodes: $(A, F), (A, E), (B, F), (B, E), (C, E), (D, F)$.

The Sum-HMM

Question 3,4,5

Interpretation of model

In this section we investigate the casino model specified in question 2.2 on the assignment. The first step was to understand the model presented. In particular, we can interpret the casino model as having 2K-tables with each table having any sort of categorical distribution on 1,...,6. A player then visits K-tables and upon each transition stays within the same K-table subgroup with probability 1/4 and switches the other K-table subgroup with a probability of 3/4. For simplicity and for the sake of analysis, we have further assumed that tables within a particular subgroup all have the same categorical distribution.

Validation of implementation

The code for the HMM-model implementation was done in python and can be found in the appendix section. To test the correctness of the implementation we used the HMM model to generate 10,000 samples of outcomes and plotted the resulting empirical distributions. In particular three tests were carried out with the following variable parameter settings:

1. All distributions equal (test 1)
   - Dice distribution 0: [1/6] * 6
   - Dice distribution 1: [1/6] * 6
2. Biased and Unbiased group (test 2)
   - Dice distribution 0: [1/6] * 6
   - Dice distribution 1: [1/12] * 4 + [2/6] * 2
3. Biased and Biased group (test 3)
   - Dice distribution 0: [2/6]*2 + [1/12]*4
   - Dice distribution 1: [1/12] * 4 + [2/6] * 2

all other parameters were held constant.

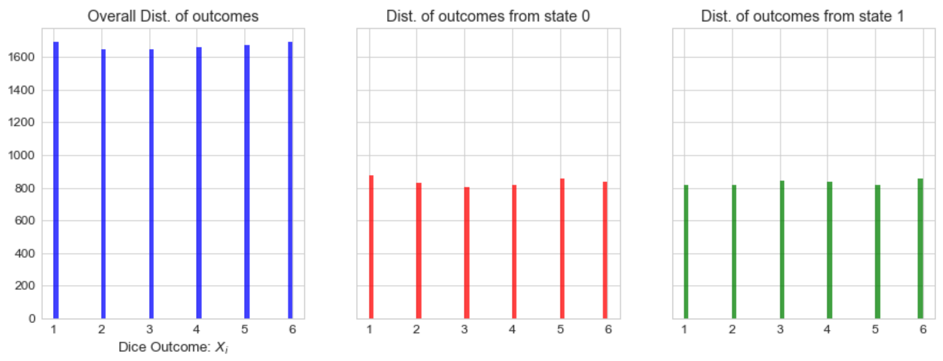

Figure 1: Resulting empirical distribution for test 1

As evident from figure 1 the resulting empirical distribution is as expected. Using only fair dice, or uniform distributions, we expect the result to be uniform.

Similarly, from both figure 7 and 8 the resulting empirical distribution and its component-wise distributions for outcomes in state 0 and outcomes in state 1 follow the expected distributions as supplied by the input parameters. This validates the correctness of the implementation.

Question 6,7

Algorithm

Notation:

- O_k: An observation, possibly hidden
- X_k: An actual outcome
- Z_k: The state of node k
- S: The observed sum of actual outcomes, $X_{1:T}$, of a particular player

The quantity of interest, for each player respectively, is $p(X_k, Z_k|S, O_{1:T})$. In order to calculate this quantity we can make use of Bayes rule.

$$p(X_k, Z_k|S, O_{1:k}) \propto p(X_k, Z_k, S, O_{1:k})$$
$$= p(O_{1:k-1}, Z_k) * p(O_k, X_k|Z_k) * p(O_{k+1:T}|Z_k) * p(S|Z_k, X_k, O_{1:T}).$$

We can then recognize the standard dynamic programming solutions. Forward $= p(O_{1:k-1}, Z_k)$, backward $= p(O_{k+1:T}|Z_k)$ and emission $= p(O_k, X_k|Z_k)$. All of these values can easily be computed in polynomial time by following the standard results derived by Bishop [1]. The new quantity of interest is therefore the probability of the sum given the state and observations. In order to be able to compute this quantity we make use of dynamic programming for sums and the key insight that we can associate a partial sum with each node in the HMM that ranges between k and 6k. We can present the algorithm with pseudo-code and mathematical notation.

Algorithm 1 DP for Sum

```
1: procedure PROBABILITY OF SUM
2:     for each node k do
3:         for each state $Z_k$ do
4:             for each partial sum $S_i$ do
5:                 $\sum_{z_{k-1}} \sum_{k-1<I<6(k-1)} \left[p(X_k = S_i - I|Z_k)p(\text{previous sum} = I|Z_{k-1})p(Z_k|Z_{k-1})\right]$
6:             end for
7:         end for
8:     end for
```

As depicted in algorithm 1, the algorithm visits each node and compute the partial sums associated with each state, Z_k. This partial sum is then propagated forward so that when computing the partial sum for $node_k$, it references the already computed partial sums from the previous node, $node_{k-1}$. Since each node has partial sums between k and 6k, hence the previous node has a partial sum between (k-1) and 6(k-1), and because there are S states for each node we have a computational complexity of $\sim O(s^2k^2)$ per node. If there are a total of k nodes in the HMM, then we will have a complete complexity of $\sim O(S^2k^3)$, which is polynomial in the number of nodes.

In the above algorithm we also supply a starting condition, which is simply the standard evidence matrix adjusted for the initial transition probabilities. Finally, in order to account for the given information, such as the given observations, $O_{1:T}$, or states, Z_k, we simply force the algorithm to take particular paths. For example, if it is given that $Z_3 = 1$, then we fix the corresponding transition probabilities: $p(Z_3 = 1|Z_2) = 1, p(Z_3 = 0|Z_2) = 0$. This has the effect of discarding any probability path that has passed through $Z_3 = 0$. It is easy to follow a similar approach with regards to the given observations.

Validation of implementation

In order to validate the implementation of the algorithm to calculate the conditional probability, two validation executions were carried out. For each test, we specified the parameters of the HMM model, ran the standard implementation to generate a set of observations and then used the algorithm to calculate the conditional probabilities of Z_k and X_k for each node. The tests can be summarized as follows:

1. Test 1
   - Dice distribution 0: [90/100] + [2/100]*5
   - Dice distribution 1: [2.5/100]*4 + [45/100]*2
   - Probability to observe: 1.0
   - Number of nodes: 4
2. Test 2
   - Dice distribution 0: [90/100] + [2/100]*5
   - Dice distribution 1: [2.5/100]*4 + [45/100]*2
   - Probability to observe: 0.4
   - Number of nodes: 4

Also, for test one the following observations were made:

- State Sequence: 0 - 1 - 0 - 0
- Observed Outcomes: 1 - 5 - 1 - 1
- Actual Outcomes: 1 - 5 - 1 - 1
- Sum of outcomes: 8

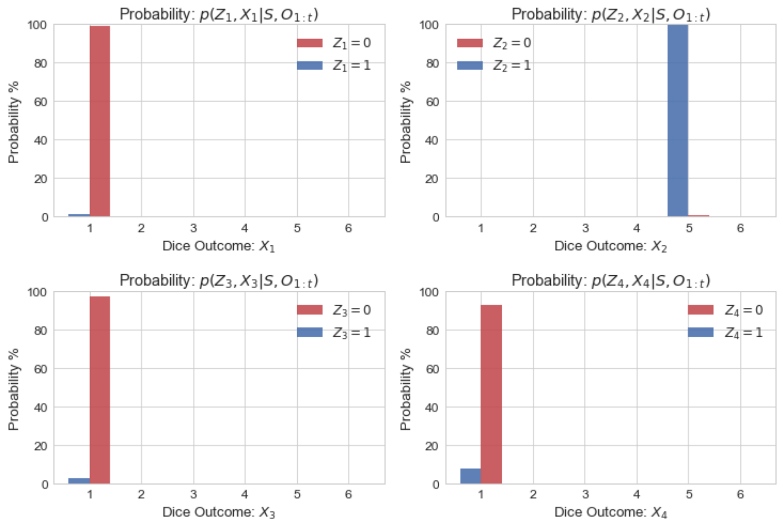

Figure 2: Conditional probabilities, test 1

The results of test one are depicted above in figure 2. We notice that for each node, representing a single time step, the conditional distributions are centered upon single outcomes. This makes sense because all the outcomes were observed and hence is known, there is no uncertainty regarding observed outcomes. There is still, however, some uncertainty regarding the actual state, Z_k, for each node since this information was not given. Also the results show the distribution is heavily weighted towards $Z_k = 0$ for nodes one, three and four and towards $Z_k = 1$ for node two. This makes sense because the original dice distributions specified meant that it is much more likely to generate an outcome of 1 from state 0 than compared to state 1. State 1, however, is much more likely to generate outcomes of 5 or 6. Finally, it is interesting to observe that the probability of $Z_k = 1$ at node four is larger than for nodes one and two. This is correctly so as the transition probabilities make it much more likely to change state than to stay in the same state when passing from one node to another.

- State Sequence: 0 - 1 - 1 - 0
- Observed Outcomes: (-1) - (-1) - 5 - 1
- Actual Outcomes: 5 - 6 - 5 - 1
- Sum of outcomes: 17

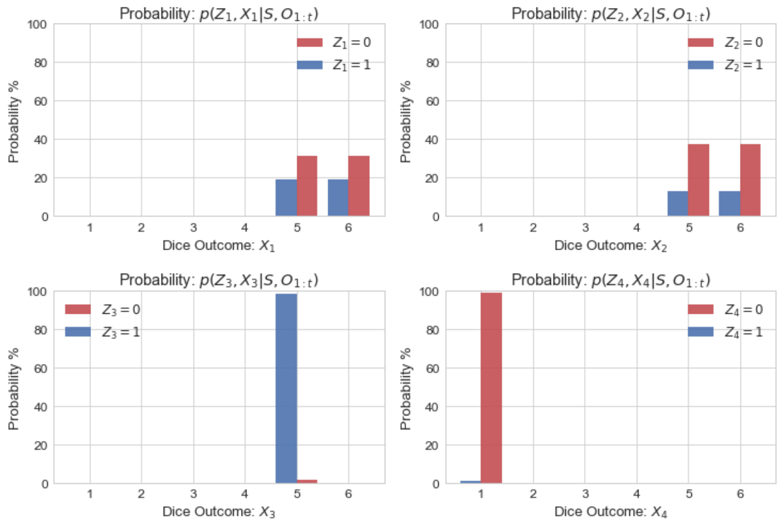

Figure 3: Conditional probabilities, test 2

Similarly as for test one, the results for test two are depicted in figure 3. The difference here is with respect to the probability of observing the outcomes, which in test two is lower. An unobserved outcome is depicted by a negative one. Again for the observed outcomes, the probability is focused on a single number. For the outcomes that were unobserved the distributions are centered upon numbers five and six. This is intuitive since knowing that the sum of all outcomes is 17, there are only two possible combinations of numbers that can generate the sum of the unknowns (17 - 6 = 11). It is also interesting to observe that the distributions for nodes one and two are fairly similar. At first this is not intuitive, but when considering that each of these conditional probabilities are calculated independently from one another it is understandable. Particularly, the information given is only the observed outcomes and its sum, hence the conditional probability expressed in node two does not consider the conditional probability of node one. Also with the lack of information, with more nodes being unobserved, the probability mass is also more spread out, again something that is to be expected.

Simple VI

Question 8,9

In this question we want to examine variational inference as a technique for approximating distributions. In particular, we will be focusing on the posterior distribution for the mean μ and precision τ of a univariate Gaussian distribution. We use the following conjugate prior for the parameters:

$$p(\mu|\tau) = \mathcal{N}(\mu_0, (\lambda_0\tau)^{-1})$$
$$p(\tau) = Gam(\alpha_0, \beta_0).$$

As these priors represent conjugate priors for the normal distribution, we can find the exact posterior analytically. This will also be of a Gamma-Normal form. The exact posterior can be described as following:

$$\alpha_n = \alpha_0 + n/2 - 1/2$$

$$\beta_n = \beta_0 + n/2 * (var_x + \frac{\lambda_0 * (\bar{x} - \mu_0)^2}{\lambda_0 + n})$$
$$\lambda_n = \lambda_0 + n$$
$$\mu_n = \frac{\mu_0 * \lambda_0 + n * \bar{x}}{\lambda_0 + n}$$
$$p(\mu|\tau) = \mathcal{N}(\mu_n, (\lambda_n \tau)^{-1})$$
$$p(\tau) = Gam(\alpha_n, \beta_n).$$

Question 10

For the following question we randomly generated a dataset consisting of 12 observations from a normal distribution with mean zero and variance two. We then initialized the variational inference algorithm by setting the initial parameters and ran the algorithm to obtain an approximate posterior distribution.

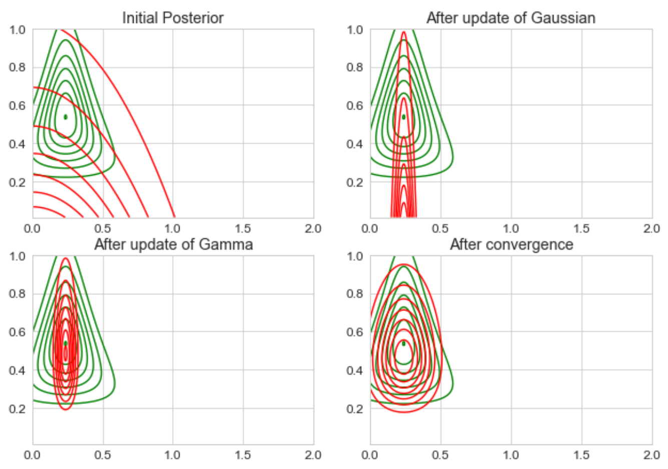

Figure 4: Variational inference on posterior parameter space. Actual posterior (green), VI posterior (red)

Figure 4 shows how the approximate distributed converges towards the true posterior distribution of the parameters as the variational inference algorithm updates each set of parameters in turn. In the above example we made use of the simplifying assumption that the posterior distribution factorizes into two independent distributions, that is to say $q(\mu, \tau) = q_\mu(\mu) q_\tau(\tau)$. Hence it will not be able to capture the full posterior distribution structure in which the joint distribution does not factorize into two independent distributions.

Sampling tables for the Sum-HMM

Question 11

To calculate the conditional probability of a sequence of states given an observed sum and an outcome sequence we can simply use Bayes rule.

$$\begin{aligned} p(Z_{1:k}|S, O_{1:k}) &\propto p(Z_{1:k}, S, O_{1:k}) \\ &= p(Z_{1:k}, O_{1:k}) p(S|O_{1:k}, Z_{1:k}) \end{aligned}$$

Looking back to question six and seven, we have already derived the appropriate algorithms to calculate these quantities in polynomial time. The question that remains is how to calculate the normalizing coefficient, $p(S, O_{1:k}$. However this is easiest done, and in polynomial time, by marginilizing over Z_k. More directly, we choose any node and then calculate the following quantity:

$$p(S, O_{1:k}) = \sum_{Z_k} p(Z_k, S, O_{1:k}) = \sum_{Z_k} p(Z_k, O_{1:k}) p(S|Z_k, O_{1:k}).$$

Again, we have already derived efficient algorithms to calculate the above quantities as these follow the same algorithm from questions six and seven.

Question 12

To sample the most likely state sequence a posteriori we consider two quantities of interest.

$$p(Z_k|S, O_{1:k}) = \frac{p(Z_k, S, O_{1:k})}{p(S, O_{1:k})} = \frac{p(Z_k, O_{1:k}) p(S|Z_k, O_{1:k})}{\sum_{Z_k} p(Z_k, O_{1:k}) p(S|Z_k, O_{1:k})}$$

and

$$p(Z_{k-1}|Z_k, S, O_{1:k}) = \frac{p(Z_{k-1}, Z_k, S, O_{1:k}}{p(Z_k, S, O_{1:k})} = \frac{p(Z_{k-1}, O_{1:k-1}) p(O_k, Z_k|Z_{k-1}) p(S|Z_{k-1:k}, O_{1:k})}{\sum_{Z_{k-1}} p(Z_{k-1}, Z_k, S, O_{1:k})}$$

where we can calculate $p(O_k, Z_k|Z_{k-1})$ as $p(O_k|Z_k)p(Z_k|Z_{k-1})$. Again, the algorithms associated with the above equations are the same ones as the ones used in question six and seven. To then sample a state sequence we can start at the end of the Markov chain, sample state Z_k and then work our way backwards, making sure to include the newly sampled state into our set of known information.

Validation of implementation

Test 1

- Dice distribution 0: [90/100] + [2/100]*5
- Dice distribution 1: [2.5/100]*4 + [45/100]*2
- Probability to observe: 1.0
- Number of nodes: 5
- State Sequence: 0 - 1 - 0 - 1 - 1
- Observed Outcomes: 1, 5, 1, 5, 5
- Actual Outcomes: 1, 5, 1, 5, 5
- Sum of outcomes: 17

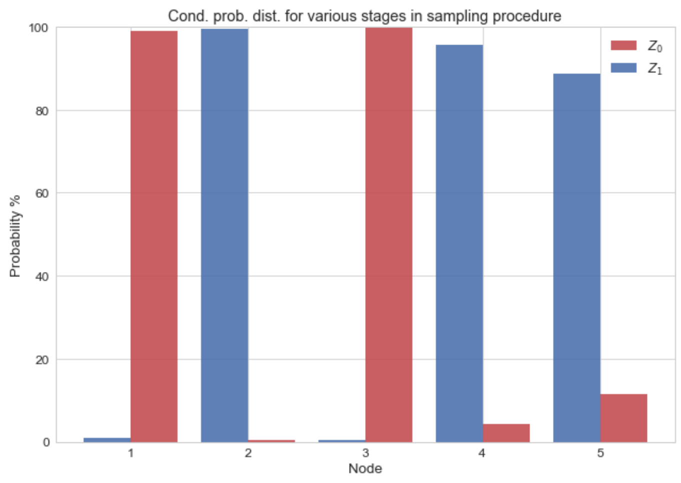

Figure 5: Sampling of state sequence

The probability distributions for each node is shown in figure 5. If we choose to sample according to maximum likelihood, then the sampled sequence generated would be: 0,1,0,1,1. Comparing with the actual state sequence that generated the results we see that it matches perfectly. This is the result to be expected as this particular test allowed for each outcome to be observed, and since the different dice distributions are heavily skewed, it is not surprising that the algorithm was able to find the exact sequence.

Test 2

- Dice distribution 0: [90/100] + [2/100]*5
- Dice distribution 1: [2.5/100]*4 + [45/100]*2
- Probability to observe: 0.5
- Number of nodes: 5
- State Sequence: 1, 0, 1, 1, 0
- Observed Outcomes: -1, 1, 6, 5, -1
- Actual Outcomes: 6, 1, 6, 5, 1
- Sum of outcomes: 19

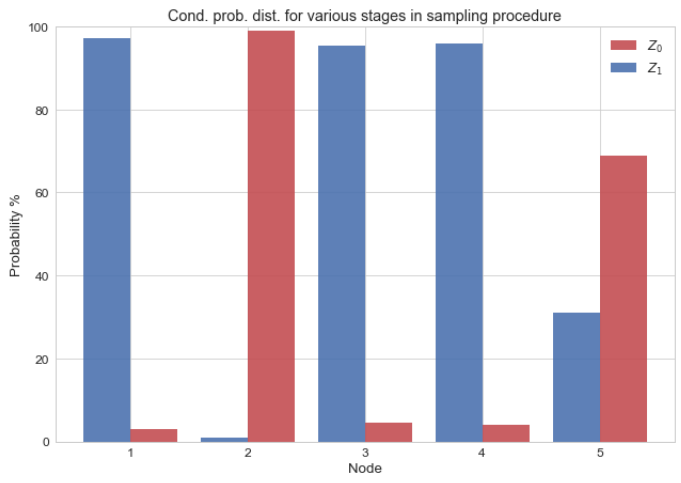

Figure 6: Sampling of state sequence

Similarly, the results for test two is shown in figure 6. Here the most likely sequence of states is 1,0,1,1,0. This shows that we were able to obtain the the actual state sequence even though we had incomplete information, two of the observations were unknown. This does, however, not necessarily have to be the case and an incorrect state sequence generation may occur for other examples. In particular, examining the distribution for node five we see that there is a significant uncertainty regarding the state than compared to other states. This comes from the fact that the actual observation was hidden and that when sampling, we start from the last node, node five. This means we must sample based only on the incomplete observations and the observed sum. Upon sampling the state for the other nodes, inference is more certain as we may base the choice on the states that were sampled before a particular node.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

Appendix A

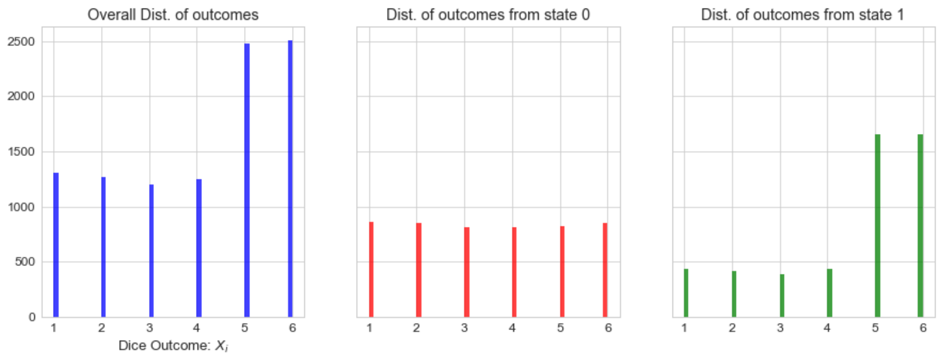

Figure 7: Resulting empirical distribution for test 2

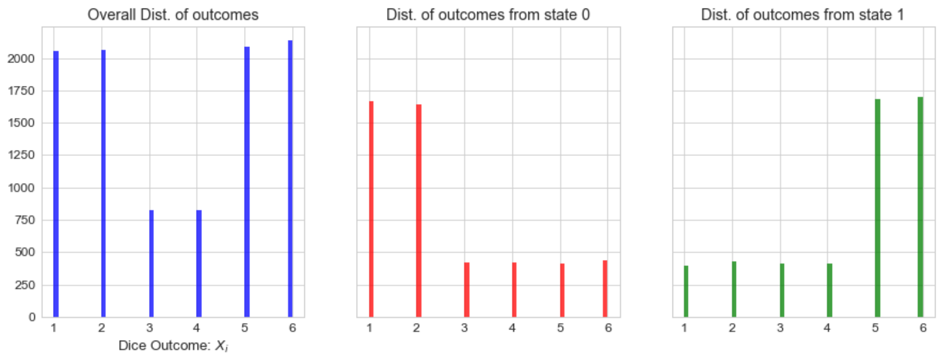

Figure 8: Resulting empirical distribution for test 3