Deep Structured Learning (IST, Fall 2020)

Homework 3

Instructor: André Martins
TA: Marcos Treviso

Deadline: Friday, December 9, 2020.

Please turn in the answers to the questions below together with the code you implemented to solve them (when applicable). Please email your solutions in **electronic format** (a single zip file) with the subject "Homework 3" to:

`deep-structured-learning-instructors@googlegroups.com`

Hard copies will not be accepted.

Question 1

Transliteration. Transliteration is the problem of converting text (usually entity names) from one script to another. For example, `васильевич` in Russian (Cyrillic script) is transliterated as `Vassiljevitch` in English (Latin script). We can regard this is as a sequence-to-sequence problem, where the sizes of the two sequences do not necessarily match.

In this exercise, we will use the Arabic-English transliteration data released by Google (`https://github.com/googlei18n/transliteration`).

Run the following commands to download the train, validation, and test partitions (resp. 12877, 1431, and 1590 word pairs):

```
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-train.txt
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-eval.txt
wget https://raw.githubusercontent.com/googlei18n/transliteration/master/ar2en-test.txt
```

1. You are going to implement a sequence-to-sequence model for this task. The input and output should respectively be an Arabic and a English word, represented left-to-right as a sequence of characters. The evaluation metric is Word Accuracy (which counts the fraction of words that were fully transliterated correctly).
   (a) (10 points) Start by determining the source and target vocabularies (don't forget to include special symbols, such as START, STOP, UNK, and PAD (if you pad). What are the vocabulary sizes?
   (b) (20 points) Implement a vanilla sequence-to-sequence model using an encoder-decoder architecture with two unidirectional LSTMs (one encoder LSTM and one decoder LSTM). Report the validation accuracy as a function of the epoch number and the final test accuracy. Hint: if you're using Pytorch, use the function `nn.LSTM` for this exercise.
   (c) (10 points) Repeat the previous exercise reverting the source string.

(d) (10 points) Turn the encoder into a bidirectional LSTM and add an attention mechanism to the decoder. Report the validation accuracy as a function of the epoch number and the final test accuracy.

Question 2

Graphical models, Simpson's paradox, and reverse engineering of a factor graph.

1. Consider the Bayesian network in Figure 1.

   (a) (10 points) Which of the conditional independence relations below are entailed by D-separation in this graph?

   1. $T \perp\!\!\!\perp Y$
   2. $T \perp\!\!\!\perp Y \mid M_1$
   3. $T \perp\!\!\!\perp Y \mid M_1, W_3$
   4. $T \perp\!\!\!\perp Y \mid M_2, W_3$
   5. $T \perp\!\!\!\perp Y \mid M_2, W_2$
   6. $T \perp\!\!\!\perp Y \mid M_1, M_2, W_3$
   7. $T \perp\!\!\!\perp Y \mid M_1, M_2, W_3, X_1$
   8. $T \perp\!\!\!\perp Y \mid M_1, M_2, W_3, X_3$
   9. $T \perp\!\!\!\perp Y \mid M_1, M_2, W_3, X_4$
   10. $T \perp\!\!\!\perp Y \mid M_1, M_2, W_3, X_1, X_4$

   (b) (10 points) Draw the corresponding Markov network and give an example of an independence relation that is lost in the conversion process.

2. Simpson's paradox is a very famous "paradox" in statistics. In this exercise you will see an example where it manifests and you will see how causal modeling eliminates the paradox. Assume that a new treatment (T) for COVID-19 is being tested on a population which is split according to age (A). We assume that both variables are binary: the value $T = 1$ indicates that the treatment is prescribed and $T = 0$ that it is not; and $A = 1$ indicates that the patient is more than 60 years old. There are two possible outcomes: either the patient survives ($Y = 1$) or dies ($Y = 0$). We want to assess the effect of prescribing the treatment to the patient's chance of surviving. The treatment has some risky side effects, therefore doctors may choose not to prescribe it to some of the older patients, which makes A a confounder (it affects both treatment and outcome). We assume the causal graph illustrated in Figure 2.

   After performing some tests, it was observed that 50 aged ($A = 1$) and 500 non-aged patients ($A = 0$) took the treatment, of which 5 aged patients and 100 non-aged ones died. On the other hand, 1400 aged ($A = 1$) and 100 non-aged patients ($A = 0$) did not take the treatment – from those, 210 aged patients and 30 non-aged ones died.

   (a) (10 points) Compute the empirical estimates $\Pr\{Y = 1 \mid T = t, A = 1\}$, $\Pr\{Y = 1 \mid T = t, A = 0\}$, and $\Pr\{Y = 1 \mid T = t\}$ for $t \in \{0, 1\}$. Why does this seem to lead to a paradox?

   (b) (10 points) Draw the causal graph when we intervene on the treatment variable. Compute $\Pr\{Y = 1 \mid do(T = t)\}$ and compare it to $\Pr\{Y = 1 \mid T = t\}$. Would you recommend prescribing the treatment? Comment on the results.

3. (10 points) Consider a factor graph with three binary variables X_1, X_2, X_3 connected to one hard constraint factor f whose compatibility function imposes a OR function, i.e.,

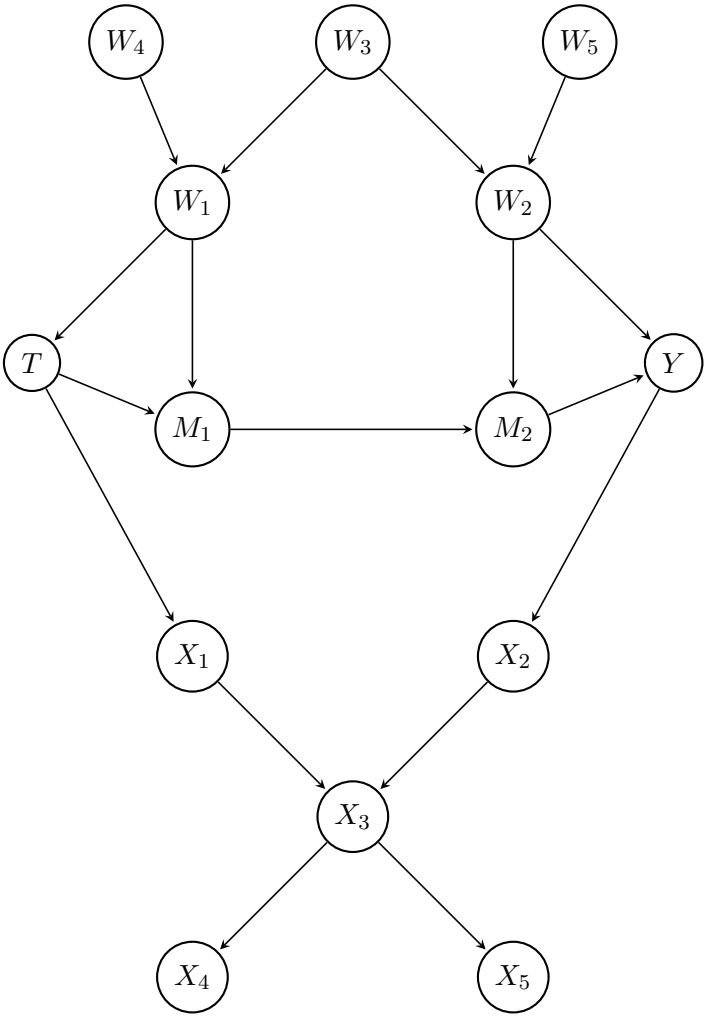

Figure 1: Example of a Bayesian network.

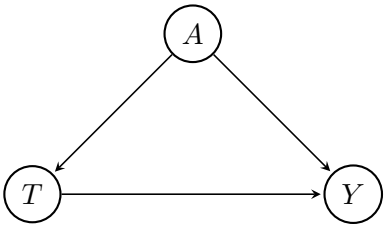

Figure 2: Example of a causal graph.

$f(x_1, x_2, x_3) = 0$ if $x_1 = x_2 = x_3 = 0$, and 1 otherwise. Each of the three variables has a unary potential function $g_i(x_i)$ for $i \in \{1, 2, 3\}$, hence the factor graph defines the distribution

$$\Pr\{X_1 = x_1, X_2 = x_2, X_3 = x_3\} \propto f(x_1, x_2, x_3) g_1(x_1) g_2(x_2) g_3(x_3).$$

We want to reverse engineer these unary potentials. We are told that the marginal probabilities of these variables according to distribution above are $\Pr\{X_1 = 1\} = 0.3$, $\Pr\{X_2 = 1\} = 0.4$, and $\Pr\{X_3 = 1\} = 0.7$. Determine the unary potential functions $g_1(x_1)$, $g_2(x_2)$, $g_3(x_3)$ up to a scale factor.