Exercise 2

Elements of Machine Learning, 2020, by Jens Petersen

The assignments in EML must be completed individually and written individually in English. Group discussions are allowed and encouraged, but in such cases you should list the group members in your handin. Your handin should include your solution to the exercises in a single pdf (not zipped), including code where relevant.

Exercises

From the textbooks (40 points)

Solve the following exercises

- PRML[1] 8.16, 8.27, ESL[2] 8.4, 10.1

Graphical Models (30 points)

A standard example in graphical model literature is the student network (Figure 1.

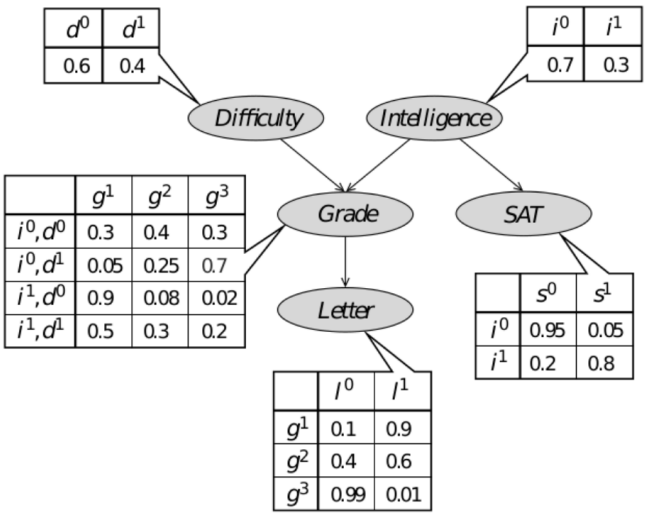

Figure 1: The student network. The model describes the relationship between class difficulty (0 = low, 1 = high), student intelligence (0 = not intelligent, 1 = intelligent), the grade the student gets in class (1 = good, 2 = average, 3 = bad), the student's score in the SAT exam (0 = low, 1 = high), and the recommendation letter the student gets from the professor (0 = not a good letter, 1 = good letter).

[1]Pattern Recognition and Machine Learning, Christopher M. Bishop

[2]Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani, and Jerome Friedman

1. What is $p(\text{Intelligence} = 1 | \text{Letter} = 1, \text{SAT} = 1)$?
2. Convert the directed graph in Figure 1 to a factor graph
3. With the Letter node as root, list the Sum-Product and Max-Sum messages in the forward and backward passes, what are the marginal probabilities and most likely configuration of each node?

Combining multiple learners (30 points)

The goal of this exercise is to predict whether an email is spam or not using various ways of combining multiple learners. For this you will need the spam dataset described in ESL, which can be downloaded from `https://archive.ics.uci.edu/ml/datasets/spambase`. We will use classification trees as the base learner.

1. Split the dataset into non-overlapping training, validation, and test sets. Motivate your choice of splitting ratios, given the exercises below.
2. Implement, train and compare results of classifying spam
   (a) using a basic classification tree method. You may use the decision tree implementation in scikit-learn for this `https://scikit-learn.org/stable/modules/tree.html`.
   (b) using bagging.
   (c) using a boosting algorithm, such as AdaBoost.M1.

   You are expected to describe, motivate, and compare possible methodological choices, including how you arrive at values for hyper parameters such as the number of iterations of AdaBoost, the number of base learners and the tree depth for the base learners. The bagging and boosting part of the exercise should be implemented by yourself, that is, it should not rely on existing implementations such as scikit-learn's ensemble methods `https://scikit-learn.org/stable/modules/ensemble.html`, however, you may use the base learner from 2a. Write up a comparison of the three approaches, what are your conclusions?