Exercise 3

Synthesis for Interpretable Machine Learning

Reliable and Interpretable Artificial Intelligence 2017
ETH Zürich

October 11, 2017

Recall from the lecture that the presented method for creating a new probabilistic model can, at a high level, be described using the following four steps:

- Pick a structure of interest.
- Define a domain specific language (DSL) for expressing functions.
- Synthesise $f_{best} \in DSL$ from dataset $\mathcal{D}$.
- Use f_{best} to compute context and make predictions.

Problem 1. In this exercise our goal is to build a probabilitic model but over sequences instead of tree structures. In particular, we will build character level language model. Consider a sample set of queries a), b), c) as shown below:

(a) Mg12 He0 Ai31 Fe14 Mg13 Ag22 **?**

(b) Mg12 He0 Ai31 Fe14 Mg13 Ag22 Fe**?**

(c) Mg12 He0 Ai31 Fe14 Mg13 Ag22 Fe15**?**

where we use symbol **?** to denote a character to be predicted by the model. You can assume that the dataset for this task consists of sentences that look similar to those shown above.

1. Design domain specific language (called TSeq) that works over sequences instead of trees.

2. Write down programs in TSeq that you believe are useful for the given queries. Are there any interesting programs that TSeq cannot express or could express more efficiently?

3. (optional) Is it possible use sequence model to model trees? If yes, describe how it can be achieved. If not, provide an example tree that cannot be encoded as a sequence.

Problem 2. In this task we will use a program from TSeq language to build a probabilistic model. Consider the following dataset which for simplicity consists of only a single sentence:

Mg12 He0 Ai31 Fe14 Mg13 Ag22 Fe15

1. We start by selecting a program $f \in TSeq$ that will parametrize the probabilistic model. For simplicity, assume we selected a function f_{bigram} that coressponds to a bigram model[1].

2. For each word in the dataset the program f_{bigram} specifies the context to be used for its prediction. Write down a table that contains the context (i.e., evaluates program f_{bigram}) for each position of query from task 1b).

3. Once we computed the context for each position we would like to use it for the actual prediction. Build a model based on maximum likelihood estimation that is used to calculate probability of character at a given position t given it context (i.e., $p(x_t \mid f_{bigram}(x_{<t}))$) as follows:

$$p(w_i \mid f_{bigram}(x_{<t})) = \frac{c(f_{bigram}(x_{<t}) \cdot w_i)}{c(f_{bigram}(x_{<t}))}$$

   where $c(\boldsymbol{v})$ denotes the number of times the value $\boldsymbol{v}$ has been seen in the training data.

4. Now consider a different program $f' \in TSeq$. In order to compare the programs f_{bigram} and f' we need to define a cost function $cost : TSeq \times \mathcal{D} \to \mathbb{R}$ that assigns cost to each program when trained on a dataset $\mathcal{D}$. Define what a suitable cost function.

[1] Bigram model conditions prediction of word x_t on a single preceding word x_{t-1}, that is, the probability $p(x_t \mid x_1 \cdots x_{t-1})$ is approximated as $p(x_t \mid x_{t-1})$.