Hidden Markov Models for Sequence Tagging

Ensimag NLP course

February 2023

The aim of this lab session is to develop an HMM sequence tagger and evaluate it on classical sequence tagging tasks, such as **part-of-speech (POS) tagging** or **Name Entity Recognition**.

Sequence tagging is a structured prediction problem consisting in assigning a sequence of n tags $\mathbf{c} = c_1, c_2, \ldots, c_n$ to an input sequence of the same length $\mathbf{w} = w_1, w_2, \ldots, w_n$.

For exemple, in POS tagging, the French sentence `"Le chat mange une pomme ."` should be assigned the POS sequence: `"D N V D N PONCT"`.

1 Hidden Markov Model

The HMM models the joint distribution $P(\mathbf{W} = \mathbf{w}, \mathbf{C} = \mathbf{c})$, where $\mathbf{w} = w_1, w_2, \ldots, w_n$ is a sequence of **observations**, $\mathbf{c} = c_1, c_2, \ldots, c_n$ is a sequence of **hidden states** of the same length.

The inference consists in finding the tag sequence with the highest probability conditioned on the input:

$$\begin{aligned}
\hat{\mathbf{c}} &= \underset{\mathbf{c} \in \mathrm{GEN}(\mathbf{w})}{\arg\max}\; P(\mathbf{C} = \mathbf{c} | \mathbf{W} = \mathbf{w}) && (1) \\
&= \underset{\mathbf{c} \in \mathrm{GEN}(\mathbf{w})}{\arg\max}\; \frac{P(\mathbf{W} = \mathbf{w} | \mathbf{C} = \mathbf{c}) \cdot P(\mathbf{C} = \mathbf{c})}{P(\mathbf{W} = \mathbf{w})} && \text{(Bayes Theorem)} \quad (2) \\
&= \underset{\mathbf{c} \in \mathrm{GEN}(\mathbf{w})}{\arg\max}\; P(\mathbf{W} = \mathbf{w} | \mathbf{C} = \mathbf{c}) \cdot P(\mathbf{C} = \mathbf{c}) && (P(\mathbf{W} = \mathbf{w}) \text{ is constant}) \quad (3)
\end{aligned}$$

where $GEN(\mathbf{w})$ is the set of every possible sequence of tags of length n.

- $P(\mathbf{W} = \mathbf{w} | \mathbf{C} = \mathbf{c})$ is the probability of the sentence when the sequence of hidden states $\mathbf{c}$ is observed (**likelihood**).
- $P(\mathbf{C} = \mathbf{c})$ is the **prior** probability of the tag sequence $\mathbf{c}$.

In order to make inference tractable, we need 2 independence assumptions.

- Order-1 Markov assumption: the hidden state at timestep t only depends on the hidden state at time step $t - 1$.
$$P(c_i | c_1, \ldots c_{i-1}) = P(c_i | c_{i-1})$$
- Emission independence: the probability of a token w_t depends only on the tag c_t:
$$P(w_i | c_1, \ldots c_n, w_1, \ldots w_{i-1}) = P(w_i | c_i)$$

Questions

1. Based on these two independence assumptions, show that
$$P(c_1, c_2, \ldots, c_n) = \prod_{i=1}^{n} P(c_i | c_{i-1}) \quad (4)$$
$$P(w_1, w_2, \ldots, w_n | c_1, c_2, \ldots, c_n) = \prod_{i=1}^{n} P(w_i | c_i) \quad (5)$$
2. Reformulate equation 3 accordingly
3. Give the size of the set $\mathrm{GEN}(w_1 w_2 \ldots w_n)$ as a function of n and of the size of the tag set d.

2 Parameter Estimation

Consider the following toy corpus:

```
le/D chat/N ferme/V la/D porte/N
le/D chien/N le/CL porte/V
```

1. Estimate the parameters of an order-1 HMM model on this toy corpus (relative frequency / maximum likelihood estimation). Do not forget to take into account artificial start-of-sentence and end-of-sentence symbols.

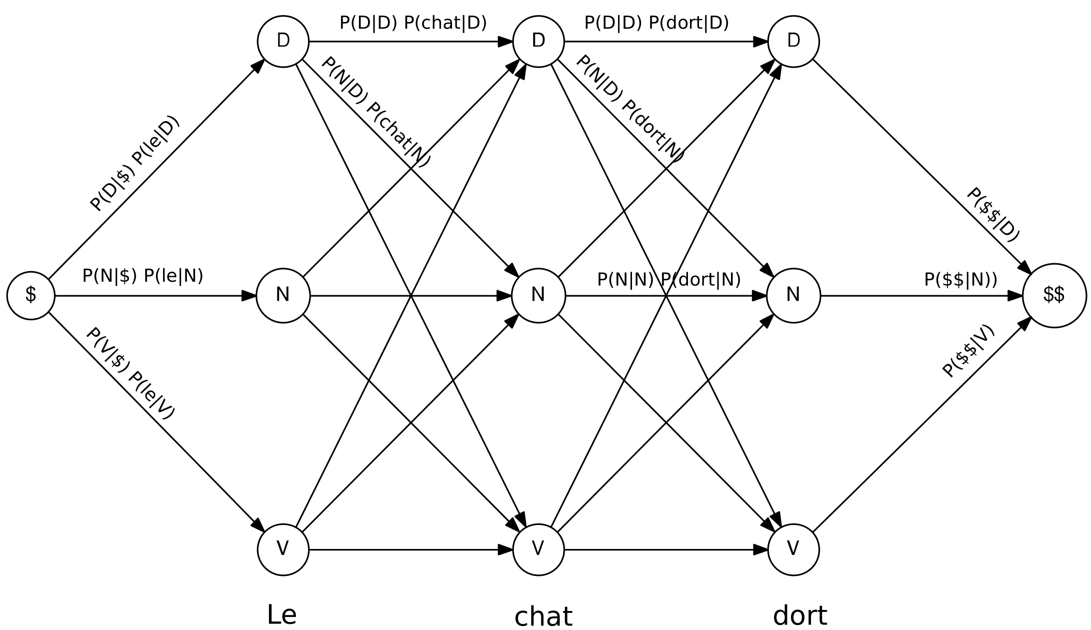
3 Inference

Inferring the best sequence of tags consists in solving the following problem:

$$\hat{\mathbf{c}} = \underset{\mathbf{c} \in GEN(\mathbf{w})}{\arg\max} \left(\prod_{i=1}^{n} P(c_i|c_{i-1}) \cdot P(w_i|c_i) \right) \cdot P(\$\$|c_n), \tag{6}$$

where $\$\$$ is the end-of-sentence symbol and $c_0 = \$$ is the start-of-sentence symbol.

This problem can be framed as a maximum-weight path search in a cycle-free oriented graph, in which each path corresponds to a possible sequence of tags and each edge is weighted based on the model's parameters. Below is an example of such a graph:



Note that depending on whether the edges are probabilities or log-probabilities, the weight of a path should be computed by multiplying or summing the weights of edges.

Consider now the following parameters:

Transitions	début	D	N	V
D	0.3	0.0	0.2	0.3
N	0.4	1.0	0.3	0.3
V	0.3	0.0	0.4	0.2
fin	0	0.0	0.1	0.2

Emissions	D	N	V
le	0.8	0.3	0.1
chat	0.1	0.4	0.4
dort	0.1	0.3	0.5

1. Compute the following probabilities:
   - $P(\mathbf{C} = (D, N, N))$
   - $P(\mathbf{W} = (\text{le,chat,dort})|\mathbf{C} = (D, N, N))$
2. Give a mathematical formulation for $P(\mathbf{W} = (le, chat, dort))$ To do so, recall that $P(x) = \sum_y P(x, y)$ (marginalization formula). How can we interpret this probability in terms of paths in the Viterbi graph?
3. Use the Viterbi algorithm to find the best path (the best sequence of tags) in the graph above.