7.3.1 Architecture of Hidden Markov Models

One best thinks of Hidden Markov models as directed graphs whose nodes represent *hidden states* with a probability distribution of emission probabilities over characters x of a finite alphabet attached to each state, and directed edges between nodes labelled with non-zero *transition probabilities*. Denote the transition probability from state p to state q by $T(p,q)$, and the emission probability for character x in state q by $E(q,x)$. There is a distinguished *start node* q_0 (and sometimes also a distinguished *final node*).

7.3.2 Causes and Effects

The goal of a Hidden Markov model is to probabilistically generate strings of emitted characters by starting at node q_0, and then walking through nodes (hidden states) according to the transition probabilities. Separating between the inner world of hidden states and outer world of observed characters is the main source of flexibility that Hidden Markov models offer to a user. Besides this, the option of modelling local interdependencies between adjacent positions in an emitted string is the second source of model flexibility. But note that the option to model long range interdependencies is not available in Hidden Markov models. There are five basic probability distributions that play a central role in Hidden Markov model usage. Calling hidden state sequences "causes" and emitted character sequences "effects", these distributions aim to measure with which probability

- a cause occurs together with an effect (*joint probability*)
- a given cause produces effects (*conditional probability of effects, given a cause*)
- a cause occurs (*marginal probability*)
- an effect occurs (*marginal probability*)
- a cause occurs, given an effect (*conditional probability of causes, given an effect*)

Formal notation and definitions are as follows, for any state sequence $W = q_1 \dots q_n$ and character sequence $S = x_1 \dots x_n$ (we assume, for simplicity, that all occurring values in denominators are greater than zero).

$$P(W) = \prod_{i=0}^{n-1} T(q_i, q_{i+1})$$

marginal distribution of causes

$$P(S \mid W) = \prod_{i=1}^{n} E(q_i, x_i)$$

conditional distribution of effects, given cause

$$P(W,S) = P(W)P(S|W)$$

joint distribution

$$P(S) = \sum_{W} P(W,S)$$

marginal distribution of effects

$$P(W \mid S) = \frac{P(W,S)}{P(S)}$$

conditional distribution of causes, given effect

The conditional distribution of effects, given causes, describes the model. It can be estimated by simply let the model run and sample data. The marginal distribution describes a priori, that is before an observation is made, knowledge about the occurrence of causes. The conditional distribution of causes, given effects, describes a posteriori, that is after having made an observation, knowledge about the presence of a certain cause. The latter is what one is interested in. For example, observing strings, a most probable state sequence in the Hidden Markov model discussed in Sect. 3.7 might define a plausible multiple alignment of strings, or observing an DNA string, a most probable state sequence in a properly designed and trained Hidden Markov model might define which parts of the string are exons of a gene.

Conditional probabilities are related by Bayes rule:

$$P(W \mid S) = \frac{P(S \mid W)P(W)}{P(S)}.$$

Given observation sequence S, the Viterbi algorithm allows efficient computation of a most probable state sequence W, that is, state sequence W such that the following holds:

$$P(W \mid S) = \max_{W'} P(W' \mid S).$$

Whereas the definitions of the probabilities above immediately can be used for an efficient computation for the cases of $P(W,S)$, $P(S \mid W)$, and $P(W)$, the definition of $P(S)$, and thus also the definition of $P(W \mid S)$, involve a sum of exponential many terms. Using the forward variable introduced in Sect. 3.8 allows replacement of this inefficient computation by an efficient one:

$$P(S) = \sum_{q \in Q} \alpha_n(q).$$