1 Representing Markov chains

1.1 Definition

A (discrete-time) Markov chain is a stochastic process for which the joint distribution of N consecutive samples $\mathbf{z} = (z_1, \ldots, z_N)^\top$ factorizes as:

$$p(\mathbf{z}) = p(z_1) \prod_{n=2}^{N} p(z_n|z_{n-1}), \tag{1}$$

1.2 Graphical and Matrix Representation

When z_n is discrete, i.e $z_n \in \mathcal{S}$ and $|\mathcal{S}| \leq \aleph_0$ [1], the transition probabilities $p(z_n|z_{n-1})$ of the Markov chain is often represented as a graph or a matrix.

For instance, if we have $z_n \in \{a, b, c\}$, the *transition matrix* is given by:

$$\mathbf{T} = \begin{bmatrix} p(z_n = a|z_{n-1} = a) & p(z_n = b|z_{n-1} = a) & p(z_n = c|z_{n-1} = a) \\ p(z_n = a|z_{n-1} = b) & p(z_n = b|z_{n-1} = a) & p(z_n = c|z_{n-1} = b) \\ p(z_n = a|z_{n-1} = c) & p(z_n = b|z_{n-1} = c) & p(z_n = c|z_{n-1} = c) \end{bmatrix}, \tag{2}$$

and the corresponding graph representation is given in Fig. 1.

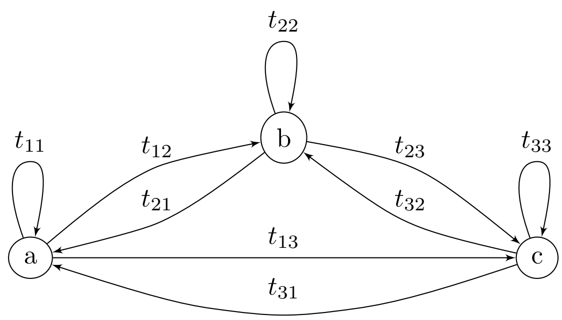

Figure 1: Graphical representation of a Markov chain. t_{ij} is the element of $\mathbf{T}$ at the ith row and jth column.

Whereas the graph conveniently represents the possible trajectories in the state-space, the transition matrix $\mathbf{T}$ allows to express state marginalization as a matrix-vector multiplication. For instance:

$$\begin{aligned} p(z_n) &= \sum_{i \in \{a,b,c\}} p(z_{n-1} = i, z_n) && (3) \\ &= \sum_{i \in \{a,b,c\}} p(z_n|z_{n-1} = i) p(z_{n-1} = i) && (4) \\ \mathbf{v}_n &= \mathbf{T}\mathbf{v}_{n-1} && (5) \end{aligned}$$

[1] This notation is a little bit pedantic but necessary in order to include Markov chains with countably infinite states (as defined in the Dirichlet-process HMM for instance).

, where:

$$\mathbf{v}_n = \begin{bmatrix} p(z_n = a) \\ p(z_n = b) \\ p(z_n = c) \end{bmatrix}. \tag{6}$$

Equation (5) is the "core" operation of many Markov chains related algorithm such as forward-backward (for training models) and viterbi (for decoding speech). For Markov chains that have a large number of states, this operation is problematic as its complexity is quadratic in the number of states: $\mathcal{O}(2D^2)$ where D is the number of states. The rest of the document describes how to exploit the structure of the Markov chains to decrease the complexity of this operation.

1.3 Compact graphical form

In many application, the transition probabilities have some structure allowing to represent the Markov chain in a more compact manner. For instance, let's consider the following transition probabilities:

$$p(z_n = j | z_{n-1} = i) = \begin{cases} \gamma + \nu_i \delta_j & \text{if } i = a \text{ and } j = b \\ \nu_i \delta_j & \text{otherwise.} \end{cases} \tag{7}$$

Defined in this way, this Markov chain has $2 \cdot 3 + 1 = 7$ parameters instead of $3 \cdot 3 = 9$ in the general case. The graphical representation of this constrained Markov chain is shown in Fig. 2.

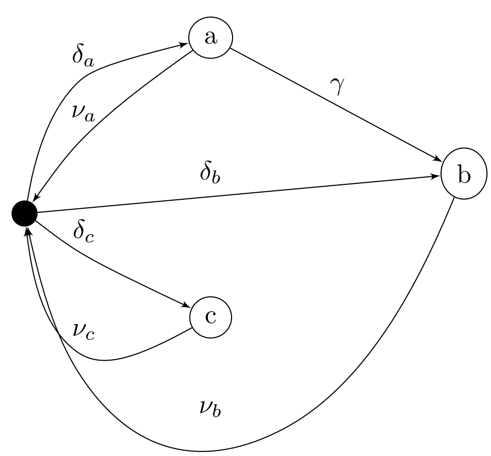

Figure 2: Graphical representation of a constrained Markov chain. The filled node is a "phony" state equivalent of the *epsilon-arc* in the WFST framework.

1.4 Efficient marginalization

In many applications, we would like to use the structure of the Markov chain to efficiently marginalize over a state. The formula in (5) can be prohibitive to evaluate if the state-space is large. The idea is to use the constraints of the Markov chain to efficiently calculate the matrix-vector product.

In our particular example, observe that the transition matrix can be written as:

$$\mathbf{T} = \mathbf{S} + \boldsymbol{\nu}\boldsymbol{\delta}^\top, \tag{8}$$

where:

$$\mathbf{S} = \begin{bmatrix} 0 & \gamma & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \; \boldsymbol{\nu} = \begin{bmatrix} \nu_a \\ \nu_b \\ \nu_c \end{bmatrix} \; \boldsymbol{\delta} = \begin{bmatrix} \delta_a \\ \delta_b \\ \delta_c \end{bmatrix}. \tag{9}$$

Consequently, we have:

$$\mathbf{T}\mathbf{v}_{n-1} = (\mathbf{S} + \boldsymbol{\nu}\boldsymbol{\delta}^\top)\mathbf{v}_{n-1}, \tag{10}$$

and using the associativity and the distributive properties of the addition and multiplication we re-write it as:

$$\mathbf{T}\mathbf{v}_{n-1} = \mathbf{S}\mathbf{v}_{n-1} + \boldsymbol{\nu}(\boldsymbol{\delta}^\top\mathbf{v}_{n-1}). \tag{11}$$

Calculating the matrix-vector product following the operation order of (11), the complexity reduces to: $\mathcal{O}(2Q+2D)$ where Q is the number of non-zero elements in $\mathbf{S}$.

Remark: in the general case, it is easy to show that:

$$\mathbf{T} = \mathbf{S} + \sum_{k}^{K} \boldsymbol{\nu}_k \boldsymbol{\delta}_k^\top, \tag{12}$$

where K is the number of "phony" states in the graphical representation of the Markov chain[2][3].

[2] Here, I assume that there is no looping path starting from a "phony" state that does not contain a "real" state. This constraint is necessarily met in practice.

[3] The factorization in (12) is not unique.