Markov Chains

Lucas Ondel

May 6, 2022

Contents

1 Semiring

A semiring $K = (\mathbb{K}, +_K, \times_K, 0_K, 1_K)$ is a set $\mathbb{K}$ equipped with a commutative binary operation $+_K$ (called "addition") with identity element 0_K and an associative binary operation $\times_K$ (called "multiplication") with identity element 1_K that distributes over $+$.

Here are commonly used semirings:

Semiring name	$\mathbb{K}$	$x +_K y$	$x \times_K y$	$x /_K y$	0_K	1_K
Boolean	$\{\text{true}, \text{false}\}$	$x \vee y$	$x \wedge y$		false	true
Log	$\mathbb{R}$	$\ln(e^x + e^y)$	$x + y$	$x - y$	$-\infty$	0
Probability	$\mathbb{R}^+$	$x + y$	$x \cdot y$	x/y	0	1
Tropical	$\mathbb{R}$	max(x, y)	$x + y$	$x - y$	$-\infty$	0
Union-Concatenation	$\{S : S \subseteq \Sigma^*\}$	$x \cup y$	$\{ab : a \in x, b \in y\}$		$\{\}$	$\{\epsilon\}$

Here are the main properties of the semirings:

Semiring name	idempotent	zero-sum free	divisible	ordered
Boolean	✓	✓		
Log		✓	✓	✓
Probability		✓	✓	✓
Tropical	✓	✓	✓	✓
Union-Concatenation	✓	✓		

2 Weighted Finite State Machine

We define a Finite State Machine (FSM) $\mathcal{M}$ by the tuple $\mathcal{M} = (Q, \Sigma, K, L, \boldsymbol{\alpha}, \mathbf{T}, \boldsymbol{\omega}, \boldsymbol{\lambda})$ where:

- $Q = \{1, \ldots, d\}$ is the set of states (with cardinality d) identified as integers
- Σ is a set of symbols
- K is an zero-sum free and ordered semiring for the FSM's weights
- L is an union-concatenation semiring defined over $\{S : S \subseteq \Sigma^*\}$ for the FSM's labels
- $\boldsymbol{\alpha} \in K^d$ is a vector such that $\alpha_i >_K 0_k$ is the initial weight of the state $i \in Q$
- $\mathbf{T} \in K^{d \times d}$ is a matrix such that T_{ij} is the transition weight from the state $i \in Q$ to the state $j \in Q$
- $\boldsymbol{\omega} \in K^d$ is a vector such that ω_i is the final weight of the state $i \in Q$
- $\boldsymbol{\lambda} \in L^d$ is a vector of symbol such that λ_i is the symbol of the state $i \in Q$

initial and final states: We say that a state i is an *initial state* if its initial weight is greater than 0, i.e $\alpha_i > 0$. Similarly, we say that a state i is a *final state* if its final weight is greater than 0, i.e. $\omega_i > 0$.

path: A path

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n), \quad \pi_i \in Q, \quad \forall i \in \{1, \ldots, n\} \tag{1}$$

is a sequence of states. The weight of a path $\boldsymbol{\pi}$ is given by the function $\mu : Q^n \to K$:

$$\mu(\boldsymbol{\pi}) = \alpha_{\pi_1} \Big[\prod_{i=2}^{n-1} T_{\pi_{i-1}, \pi_i} \Big] \omega_{\pi_N}. \tag{2}$$

Similarly, the label sequence of the path $\boldsymbol{\pi}$ is given by the function $\sigma : Q^n \to L$:

$$\sigma(\boldsymbol{\pi}) = \prod_{n=1}^{N} \lambda_{\pi_n}. \tag{3}$$

input sequence: We define an input sequence $\mathbf{s}$ to a FSM as a sequence of labels:

$$\mathbf{s} = \prod_{i=1}^{n} s_i, \quad s_i \in L. \tag{4}$$

The weight of an input sequence is given by the function $\nu : L \to K$:

$$\nu(\mathbf{s}) = \sum_{\mathbf{x} \in P} \mu(\mathbf{x}), \tag{5}$$

where $P = \{\boldsymbol{\pi} : \ \sigma(\boldsymbol{\pi}) = \mathbf{s}, \ \boldsymbol{\pi} \in \mathcal{M}\}$ is the set of paths from $\mathcal{M}$ with label sequence $\mathbf{s}$. We say that an input sequence is *accepted by* $\mathcal{M}$ if $\nu(\mathbf{s}) > 0$. We denote S the set of accepted input sequence of a FSM, i.e. $S = \{\mathbf{s} : \nu(\mathbf{s}) > 0\}$.

total sum: We define the total sum of a FSM η as the sum of all the weights of its accepted input sequence:

$$\zeta = \sum_{\mathbf{s} \in S} \nu(\mathbf{s}). \tag{6}$$

The total sum can be estimated efficiently via dynamic programming. Let be ζ_n the *partial total sum* of a FSM $\mathcal{M}$ defined as the sum of all the weights of accepted input sequence of size smaller of equal to n:

$$\zeta_n = \sum_{\mathbf{s} \in \{\mathbf{s}:\, |\mathbf{s}| \leq n,\, \mathbf{s} \in S\}} \nu(\mathbf{s}). \tag{7}$$

ζ_n can be calculated through the following recursion:

$$\mathbf{v}_n = \mathbf{T}^\top \mathbf{v}_{n-1} \tag{8}$$

$$\zeta_n = \boldsymbol{\omega}^\top \mathbf{v}_n, \tag{9}$$

where the recursion is initialized with $\mathbf{v}_1 = \boldsymbol{\alpha}$

3 FSM operations

We describe here some operations over FSMs. A FSM $\mathcal{M}_i$ is defined as:

$$\mathcal{M}_i = (Q_i, \Sigma_i, K_i, L_i, \boldsymbol{\alpha}_i, \mathbf{T}_i, \boldsymbol{\omega}_i, \boldsymbol{\lambda}_i). \tag{10}$$

The path weight, path label and input sequencs weight function associated to this FSM are denoted $\mu_i(\cdot)$, $\sigma_i(\cdot)$ and $\nu_i(\cdot)$ respectively. The set of input sequences accepted by $\mathcal{M}_i$ is $S_i = \{\mathbf{s} : \nu_i(\mathbf{s}) > 0\}$. In the following, for binary operator over 2 FSMs, we assume tacitly that both FSMs have the same set of symbols, i.e. $\Sigma = \Sigma_1 = \Sigma_2$, the same weight semiring, i.e. $K = K_1 = K_2$, and the same label semiring, i.e. $L = L_1 = L_2$.

3.1 Renormalization

We say that a FSM $\mathcal{M}$ is *normalized* if (i) the sum of its intial weights sum up to 1, i.e. $\sum_{i \in Q} \alpha_i = 1$, and (ii) the sum of the transition weights from a state i to all other states and the final weight of the state i sum up to 1, i.e. $\omega_i + \sum_{j \in Q} T_{ij} = 1$. A FSM $\mathcal{M}_1$ with a zero-sum free and divisble weight semiring can be transformed into normalized FSM $\mathcal{M}_2$ via the renormalization operation: $\mathcal{M}_2 = \text{renorm}(\mathcal{M}_1)$. The resulting FSM is obtained by the following construction:

$$Q_2 = Q_1 \qquad \boldsymbol{\alpha}_2 = \Big(\sum_i \alpha_{1i}\Big)^{-1} \boldsymbol{\alpha}_1 \tag{11}$$

$$\mathbf{T}_2 = \begin{bmatrix} (\omega_1 + \sum_i T_{1,1i})^{-1} \mathbf{T}_{1,1} \\ \vdots \\ (\omega_d + \sum_i T_{1,di})^{-1} \mathbf{T}_{1,d} \end{bmatrix} \qquad \boldsymbol{\omega}_2 = \begin{bmatrix} (\omega_1 + \sum_i T_{1,1i})^{-1} \\ \vdots \\ (\omega_d + \sum_i T_{1,di})^{-1} \end{bmatrix} \odot \boldsymbol{\omega}_1 \tag{12}$$

$$\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_1 \tag{13}$$

3.2 Union

The union of two FSMs $\mathcal{M}_1 \cup \mathcal{M}_2$ gives a FSM $\mathcal{M}_3$ such that $S_3 = S_1 \cup S_2$ and $\forall \mathbf{s} \in S_3,\ \nu_3(\mathbf{s}) = \nu_1(\mathbf{s}) + \nu_2(\mathbf{s})$. $\mathcal{M}_3$ can be obtained with the following construction:

$$Q_3 = \{1, \dots, |Q_1| + |Q_2|\} \qquad \boldsymbol{\alpha}_3 = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} \tag{14}$$

$$\mathbf{T}_3 = \begin{bmatrix} \mathbf{T}_1 & \\ & \mathbf{T}_2 \end{bmatrix} \qquad \boldsymbol{\omega}_3 = \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{bmatrix} \tag{15}$$

$$\boldsymbol{\lambda}_3 = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} \tag{16}$$

3.3 Concatenation

The concatenation of two FSMs concat$(M_1, \mathcal{M}_2)$ gives a FSM $\mathcal{M}_3$ such that $S_3 = \{\mathbf{s}_1\mathbf{s}_2 : \mathbf{s}_1 \in S_1,\ \mathbf{s}_2 \in S_2\}$. $\mathcal{M}_3$ can be obtained with the following construction:

$$Q_3 = \{1, \dots, |Q_1| + |Q_2|\} \qquad \boldsymbol{\alpha}_3 = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ 0\boldsymbol{\alpha}_2 \end{bmatrix} \tag{17}$$

$$\mathbf{T}_3 = \begin{bmatrix} \mathbf{T}_1 & \boldsymbol{\omega}_1\boldsymbol{\alpha}_2^\top \\ & \mathbf{T}_2 \end{bmatrix} \qquad \boldsymbol{\omega}_3 = \begin{bmatrix} 0\boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \end{bmatrix} \tag{18}$$

$$\boldsymbol{\lambda}_3 = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} \tag{19}$$

3.4 Reversal

The reversal (denoted $^\top$) of a FSM $\mathcal{M}_1$ yields a FSM $\mathcal{M}_2 = \mathcal{M}_1^\top$ such that $S_2 = \{\overleftarrow{\mathbf{s}} : \mathbf{s} \in S_1\}$ where $\overleftarrow{\mathbf{s}}$ is the sequence $\mathbf{s}$ in reversed order. $\mathcal{M}_2$ is obtained by the following construction:

$$Q_2 = Q_1 \qquad \boldsymbol{\alpha}_2 = \boldsymbol{\omega}_1 \tag{20}$$

$$\mathbf{T}_2 = \mathbf{T}_1^\top \qquad \boldsymbol{\omega}_2 = \boldsymbol{\alpha}_1 \tag{21}$$

$$\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_1 \tag{22}$$

3.5 Hierarchical FSM - Composition

In many applications we would like to process signals with different levels of structure. A typical example is speech where we have phonotactic structure, syllabic structure and lexical structure. To handle such phenomenon, it is necessary to build *hierarchical* FSMs, i.e. FSMs where each state is itself associated to another FSM. To build a such FSM we need to *compose* a high-level FSM with a set of sub-level FSMs.

Let's consider a FSM $\mathcal{M}_1$ with d states. We would like to compose $\mathcal{M}_1$ with a sequence of d FSMs $\mathcal{M}^{1:d} = (\mathcal{M}^1, \dots, \mathcal{M}^d)$ such that the ith state of $\mathcal{M}_1$ is associated with the FSM $\mathcal{M}^i$. The composition of $\mathcal{M}_1$ and $\mathcal{M}^{1:d}$, denoted $\mathcal{M}_1 \circ \mathcal{M}^{1:d}$, yields a FSM $\mathcal{M}_2$ which can be obtained with the following construction:

$$Q_2 = \{1, \dots, \sum_{i=1}^{d} |Q^i|\} \tag{23}$$

$$\boldsymbol{\alpha}_2 = \begin{bmatrix} \alpha_{1,1}\boldsymbol{\alpha}^1 \\ \vdots \\ \alpha_{1,d}\boldsymbol{\alpha}^d \end{bmatrix} \tag{24}$$

$$\mathbf{T}_2 = \begin{bmatrix} \mathbf{T}^1 & & \\ & \ddots & \\ & & \mathbf{T}^d \end{bmatrix} + \boldsymbol{\Omega}\mathbf{T}_1\boldsymbol{\mathcal{A}}^\top \tag{25}$$

$$\boldsymbol{\omega}_2 = \begin{bmatrix} \omega_{1,1}\boldsymbol{\omega}^1 \\ \vdots \\ \omega_{1,d}\boldsymbol{\omega}^d \end{bmatrix} \tag{26}$$

$$\boldsymbol{\lambda}_2 = \begin{bmatrix} \lambda_{1,1}\boldsymbol{\lambda}^1 \\ \vdots \\ \lambda_{1,d}\boldsymbol{\lambda}^d \end{bmatrix}, \tag{27}$$

where, $N(\cdot)$ is a function returning the non-zero elements of a matrix as a vector (ordered column-wise) and $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\Omega}$ and $\boldsymbol{\Lambda}$ are block-diagonal matrices defined as

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{\alpha}^1 & & \\ & \ddots & \\ & & \boldsymbol{\alpha}^d \end{bmatrix}, \qquad \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\omega}^1 & & \\ & \ddots & \\ & & \boldsymbol{\omega}^d \end{bmatrix}. \tag{28}$$

3.6 Determinization

$$\mathbf{s}_n = \mathbf{T}_L^\top \mathbf{s}_{n-1} \tag{29}$$

where

$$\mathbf{s}_1 = 1_L[\boldsymbol{\alpha}_1] \tag{30}$$