On Computing the Total Variation Distance of Hidden Markov Models

Stefan Kiefer
University of Oxford, United Kingdom

Abstract

We prove results on the decidability and complexity of computing the total variation distance (equivalently, the L_1-distance) of hidden Markov models (equivalently, labelled Markov chains). This distance measures the difference between the distributions on words that two hidden Markov models induce. The main results are: (1) it is undecidable whether the distance is greater than a given threshold; (2) approximation is #P-hard and in PSPACE.

2012 ACM Subject Classification Theory of computation → Probabilistic computation, Theory of computation → Random walks and Markov chains

Keywords and phrases Labelled Markov Chains, Hidden Markov Models, Distance, Decidability, Complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.130

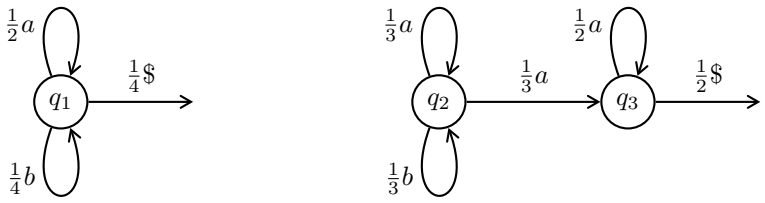
Related Version [15], https://arxiv.org/abs/1804.06170

Funding The author is supported by a Royal Society University Research Fellowship.

Acknowledgements The author thanks anonymous referees for their helpful comments.

1 Introduction

A (discrete-time, finite-state, finite-word) *labelled Markov chain (LMC)* (often called *hidden Markov model*) has a finite set Q of states and for each state a probability distribution over its outgoing transitions. Each outgoing transition is labelled with a letter from an alphabet Σ and leads to a target state, or is labelled with an end-of-word symbol \$. Here are two LMCs:



The LMC starts in a given initial state (or in a random state according to a given initial distribution), picks a random transition according to the state's distribution over the outgoing transitions, outputs the transition label, moves to the target state, and repeats until the end-of-word label \$ is emitted. This induces a probability distribution over finite words (excluding the end-of-word label \$). In the example above, if q_1 and q_2 are the initial states then the LMCs induce distributions π_1, π_2 with $\pi_1(aa) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4}$ and $\pi_2(aa) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} \cdot \frac{1}{2}$. LMCs are widely employed in fields such as speech recognition (see [23] for a tutorial), gesture recognition [4], signal processing [8], and climate modeling [1]. LMCs are heavily used in computational biology [12], more specifically in DNA modeling [6] and biological

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 130; pp. 130:1–130:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

sequence analysis [11], including protein structure prediction [17] and gene finding [2]. In computer-aided verification, LMCs are the most fundamental model for probabilistic systems; model-checking tools such as Prism [18] or Storm [9] are based on analyzing LMCs efficiently.

A fundamental yet non-trivial question about LMCs is whether two LMCs generate the same distribution on words. This problem itself has applications in verification [16] and can be solved in polynomial time using algorithms that are based on linear algebra [24, 22, 7]. If two such distributions are not equal, one may ask how different they are. There exist various *distances* between discrete distributions, see, e.g., [7, Section 3]. One of them is the total variation distance (in the following just called *distance*), which can be defined by $d(\pi_1, \pi_2) = \max_{W \subseteq \Sigma^*} |\pi_1(W) - \pi_2(W)|$ in the case of LMCs. That is, $d(\pi_1, \pi_2)$ is the largest possible difference between probabilities that π_1 and π_2 assign to the same set of words. This distance is, up to a factor 2, equal to the L_1-norm of the difference between π_1 and π_2, i.e., $2d(\pi_1, \pi_2) = \sum_{w \in \Sigma^*} |\pi_1(w) - \pi_2(w)|$. Clearly, π_1 and π_2 are equal if and only if their distance is 0.

It is immediate from the definition of the distance that if L is a family of LMCs whose pairwise distances are bounded by $b \geq 0$ then for any event $W \subseteq \Sigma^*$ and any two LMCs $\mathcal{M}_1, \mathcal{M}_2 \in L$ we have $|\pi_1(W) - \pi_2(W)| \leq b$. From a verification point of view, this means that one needs to model check only one LMC in the family to obtain an approximation within b for the probabilities that the LMCs satisfy a given property W. Therefore, computing or approximating the distance can make model checking more efficient. It is shown in [3] that the *bisimilarity pseudometric* defined in [10] is an upper bound on the total variation distance and can be computed in polynomial time. The bisimilarity pseudometric has more direct bearings on *branching-time* system properties, which, in addition to emitted labels, take LMC states into account (not considered in this paper).

The problem of computing the distance was first studied in [20]: they show that computing the distance is NP-hard. In [7] it was shown that even approximating the distance within an $\varepsilon > 0$ given in binary is NP-hard. In this paper we improve these results. We show that it is undecidable whether the distance is greater than a given threshold. Further we show that approximating the distance is #P-hard and in PSPACE. The #P-hardness construction is relatively simple, perhaps simpler than the construction underlying the NP-hardness result in [7]. In contrast, our PSPACE algorithm requires a combination of special techniques: rounding-error analysis in floating-point arithmetic and Ladner's result [19] on counting in polynomial space.

2 Preliminaries

Let Q be a finite set. We view elements of $\mathbb{R}^Q$ as *vectors*, more specifically as row vectors. We write $\mathbf{1}$ for the all-1 vector, i.e., the element of $\{1\}^Q$. For a vector $\mu \in \mathbb{R}^Q$, we denote by $\mu^\top$ its transpose, a column vector. A vector $\mu \in [0,1]^Q$ is a *distribution over* Q if $\mu \mathbf{1}^\top = 1$. For $q \in Q$ we write δ_q for the (*Dirac*) distribution over Q with $\delta_q(q) = 1$ and $\delta_q(r) = 0$ for $r \in Q \setminus \{q\}$. We view elements of $\mathbb{R}^{Q \times Q}$ as *matrices*. A matrix $M \in [0,1]^{Q \times Q}$ is called *stochastic* if each row sums up to one, i.e., $M\mathbf{1}^\top = \mathbf{1}^\top$.

▶ **Definition 1.** A *labelled (discrete-time, finite-state, finite-word) Markov chain (LMC)* is a quadruple $\mathcal{M} = (Q, \Sigma, M, \eta)$ where Q is a finite set of states, Σ is a finite alphabet of labels, the mapping $M : \Sigma \to [0,1]^{Q \times Q}$ specifies the transitions, and $\eta \in [0,1]^Q$, with $\eta^\top + \sum_{a \in \Sigma} M(a)\mathbf{1}^\top = \mathbf{1}^\top$, specifies the end-of-word probability of each state.

Intuitively, if the LMC is in state q, then with probability $M(a)(q, q')$ it emits a and moves to state q', and with probability $\eta(q)$ it stops emitting labels. For the complexity results

in this paper, we assume that all numbers in η and in the matrices $M(a)$ for $a \in \Sigma$ are rationals given as fractions of integers represented in binary. We extend M to the mapping $M : \Sigma^* \to [0,1]^{Q \times Q}$ with $M(a_1 \cdots a_k) = M(a_1) \cdots M(a_k)$ for $a_1, \ldots, a_k \in \Sigma$. Intuitively, if the LMC is in state q then with probability $M(w)(q,q')$ it emits the word $w \in \Sigma^*$ and moves (in $|w|$ steps) to state q'. We require that each state of an LMC have a positive-probability path to some state q with $\eta(q) > 0$.

Fix an LMC $\mathcal{M} = (Q, \Sigma, M, \eta)$ for the rest of this section. To an (initial) distribution π over Q we associate the discrete probability space $(\Sigma^*, 2^{\Sigma^*}, \Pr_\pi)$ with $\Pr_\pi(w) := \Pr_\pi(\{w\}) := \pi M(w) \eta^\top$. To avoid clutter and when confusion is unlikely, we may identify the distribution $\pi \in [0,1]^Q$ with its induced probability measure $\Pr_\pi$; i.e., for a word or set of words W we may write $\pi(W)$ instead of $\Pr_\pi(W)$.

Given two initial distributions π_1, π_2, the *(total variation) distance* between π_1 and π_2 is defined as follows:[1]

$$d(\pi_1, \pi_2) \;:=\; \sup_{W \subseteq \Sigma^*} |\pi_1(W) - \pi_2(W)| \,.$$

As $\pi_1(W) - \pi_2(W) = \pi_2(\Sigma^* \setminus W) - \pi_1(\Sigma^* \setminus W)$, we have $d(\pi_1, \pi_2) = \sup_{W \subseteq \Sigma^*} (\pi_1(W) - \pi_2(W))$. The following proposition follows from basic principles, see, e.g., [21, Lemma 11.1]. In particular, it says that the supremum is attained and the total variation distance is closely related to the L_1-distance:

▶ **Proposition 2.** *Let $\mathcal{M}$ be an LMC. For any two initial distributions π_1, π_2 we have:*

$$d(\pi_1, \pi_2) \;=\; \max_{W \subseteq \Sigma^*} (\pi_1(W) - \pi_2(W)) \;=\; \frac{1}{2} \sum_{w \in \Sigma^*} |\pi_1(w) - \pi_2(w)|$$

The maximum is attained by $W = \{w \in \Sigma^* : \pi_1(w) \geq \pi_2(w)\}$.

In view of this proposition, all complexity results on the (total variation) distance hold equally for the L_1-distance.

An LMC $\mathcal{M}$ is called *acyclic* if its transition graph is acyclic. Equivalently, $\mathcal{M}$ is acyclic if for all $q \in Q$ we have that $\Pr_{\delta_q}$ has finite support, i.e., $\{w \in \Sigma^* : \Pr_{\delta_q}(w) > 0\}$ is finite.

3 The Threshold-Distance Problem

In [20, Section 6] (see also [7, Theorem 7]), a reduction is given from the *clique* decision problem to show that computing the distance in LMCs is NP-hard. In that reduction the distance is rational and its bit size polynomial in the input. It was shown in [5, Proposition 12] that the distance d can be irrational. Define the *non-strict (resp. strict) threshold-distance* problem as follows: Given an LMC, two initial distributions π_1, π_2, and a threshold $\tau \in [0,1] \cap \mathbb{Q}$, decide whether $d(\pi_1, \pi_2) \geq \tau$ (resp. $d(\pi_1, \pi_2) > \tau$). In [5, Proposition 14] it was shown that the non-strict threshold-distance problem is NP-hard with respect to Turing reductions.

In the following two subsections we consider the threshold-distance problem for general and acyclic LMCs, respectively.

[1] One could analogously define the total variation distance between two LMCs $\mathcal{M}_1 = (Q_1, \Sigma, M_1, \eta_1)$ and $\mathcal{M}_2 = (Q_2, \Sigma, M_2, \eta_2)$ with initial distributions π_1 and π_2 over Q_1 and Q_2, respectively. Our definition is without loss of generality, as one can take the LMC $\mathcal{M} = (Q, \Sigma, M, \eta)$ where Q is the disjoint union of Q_1 and Q_2, and M, η are defined using M_1, M_2, η_1, η_2 in the straightforward manner.

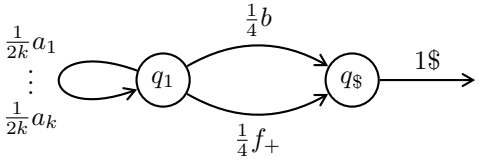
3.1 General LMCs

We show:

▶ **Theorem 3.** *The strict threshold-distance problem is undecidable.*

Proof. We reduce from the emptiness problem for probabilistic automata. A *probabilistic automaton* is a tuple $\mathcal{A} = (Q, \Sigma, M, \alpha, F)$ where Q is a finite set of states, Σ is a finite alphabet of labels, the mapping $M : \Sigma \to [0,1]^{Q\times Q}$, where $M(a)$ is a stochastic matrix for each $a \in \Sigma$, specifies the transitions, $\alpha \in [0,1]^Q$ is an initial distribution, and $F \subseteq Q$ is a set of accepting states. Extend M to $M : \Sigma^* \to [0,1]^{Q\times Q}$ as in the case of LMCs. In the case of a probabilistic automaton, $M(w)$ is a stochastic matrix for each $w \in \Sigma^*$. For each $w \in \Sigma^*$ define $\Pr_{\mathcal{A}}(w) := \alpha M(w)\eta^\top$ where $\eta \in \{0,1\}^Q$ denotes the characteristic vector of F. The probability $\Pr_{\mathcal{A}}(w)$ can be interpreted as the probability that $\mathcal{A}$ accepts w, i.e., the probability that after inputting w the automaton $\mathcal{A}$ is in an accepting state. The *emptiness problem* asks, given a probabilistic automaton $\mathcal{A}$, whether there is a word $w \in \Sigma^*$ such that $\Pr_{\mathcal{A}}(w) > \frac{1}{2}$. This problem is known to be undecidable [22, p. 190, Theorem 6.17].

In the following we assume $\Sigma = \{a_1, \ldots, a_k\}$. Given a probabilistic automaton $\mathcal{A}$ as above, construct an LMC $\mathcal{M} = (Q \cup \{q_1, q_\$\}, \Sigma \cup \{b, f_+, f_-\}, \overline{M}, \delta_{q_\$})$ such that $q_1, q_\$$ are fresh states, and b, f_+, f_- are fresh labels. The transitions originating in the fresh states $q_1, q_\$$ are as follows:

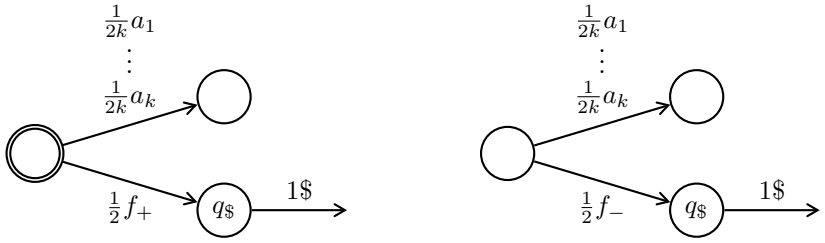


Here and in the subsequent pictures we use a convention that there be a state $q_\$$ with $\eta(q_\$) = 1$ and that $\eta(q) = 0$ hold for all other states.

Define $\pi_1 := \delta_{q_1}$. Then for all $w \in \Sigma^*$ we have:

$$\pi_1(wb) \;=\; \pi_1(wf_+) \;=\; \left(\frac{1}{2k}\right)^{|w|} \cdot \frac{1}{4} \tag{1}$$

The transitions originating in the states in Q are defined so that all $q \in Q$ emit each $a \in \Sigma$ with probability $\frac{1}{2k}$ (like q_1). For all $q \in F$ there is a transition to $q_\$$ labelled with $\frac{1}{2}$ and f_+; for all $q \in Q \setminus F$ there is a transition to $q_\$$ labelled with $\frac{1}{2}$ and f_-:



Formally, for $q, r \in Q$ and $a \in \Sigma$ set $\overline{M}(a)(q,r) := \frac{1}{2k} M(a)(q,r)$. For $q \in F$ set $\overline{M}(f_+)(q, q_\$) := \frac{1}{2}$, and for $q \in Q \setminus F$ set $\overline{M}(f_-)(q, q_\$) := \frac{1}{2}$. Define $\pi_2 := \alpha$ (in the natural way, i.e., with $\pi_2(q_1) = \pi_2(q_\$) = 0$). Then for all $w \in \Sigma^*$ we have:

$$\begin{aligned} \pi_2(wf_+) &= \left(\frac{1}{2k}\right)^{|w|} \cdot \Pr_{\mathcal{A}}(w) \cdot \frac{1}{2} \qquad \text{and} \\ \pi_2(wf_-) &= \left(\frac{1}{2k}\right)^{|w|} \cdot (1 - \Pr_{\mathcal{A}}(w)) \cdot \frac{1}{2} \end{aligned} \tag{2}$$

Consider $L := \Sigma^*\{b, f_+\}$. We have $\pi_1(L) = 1$. One can compute $\pi_2(L)$ in polynomial time by computing the probability of reaching a transition labelled by f_+ (the label b is not reachable). We claim that there is $w \in \Sigma^*$ with $\Pr_{\mathcal{A}}(w) > \frac{1}{2}$ if and only if $d(\pi_1, \pi_2) > \pi_1(L) - \pi_2(L)$. It remains to prove this claim.

Suppose there is no $w \in \Sigma^*$ with $\Pr_{\mathcal{A}}(w) > \frac{1}{2}$. Then, by (1) and (2), for all $w \in \Sigma^*$ we have $\pi_1(wf_+) \geq \pi_2(wf_+)$. Hence:

$$\{w \in (\Sigma \cup \{b, f_+, f_-\})^* : \pi_1(w) > 0,\ \pi_1(w) \geq \pi_2(w)\} \quad = \quad L$$

By Proposition 2 it follows $d(\pi_1, \pi_2) = \pi_1(L) - \pi_2(L)$.

Conversely, suppose there is $w \in \Sigma^*$ with $\Pr_{\mathcal{A}}(w) > \frac{1}{2}$. Consider $L' := L \setminus \{wf_+\}$. We have:

$$\begin{aligned} d(\pi_1, \pi_2) &\geq \pi_1(L') - \pi_2(L') && \text{Proposition 2} \\ &= \pi_1(L) - \pi_1(wf_+) - \pi_2(L) + \pi_2(wf_+) && \text{definition of } L' \\ &= \pi_1(L) - \pi_2(L) + \left(\frac{1}{2k}\right)^{|w|} \cdot \left(\frac{1}{2}\Pr_{\mathcal{A}}(w) - \frac{1}{4}\right) && \text{by (1) and (2)} \\ &> \pi_1(L) - \pi_2(L) && \Pr_{\mathcal{A}}(w) > \frac{1}{2} \end{aligned}$$

◀

Cortes, Mohri, and Rastogi [7] conjectured "that the problem of computing the [...] distance [...] is in fact undecidable", see the discussion after the proof of [7, Theorem 7]. Theorem 3 proves *one interpretation* of that conjecture. But the distance can be approximated with arbitrary precision, cf. Section 4, so the distance is "computable" in this sense.

In [5, Theorem 15] it was shown that there is a polynomial-time many-one reduction from the square-root-sum problem to the non-strict threshold-distance problem for LMCs. Decidability of the non-strict threshold-distance problem remains open.

3.2 Acyclic LMCs

It was shown in [20, Section 6] and [5, Proposition 14] that the non-strict threshold-distance problem is NP-hard with respect to Turing reductions, even for acyclic LMCs. We improve this result to PP-hardness:

▶ **Proposition 4.** *The non-strict and strict threshold-distance problems are PP-hard, even for acyclic LMCs and even with respect to many-one reductions.*

The proof uses the connection between PP and #P. Consider the problem #NFA, which is defined as follows: given a nondeterministic finite automaton (NFA) $\mathcal{A}$ over alphabet Σ, and a number $n \in \mathbb{N}$ in unary, compute $|L(\mathcal{A}) \cap \Sigma^n|$, i.e., the number of accepted words of length n. The problem #NFA is #P-complete [14]. The following lemma forms the core of the proof of Proposition 4:

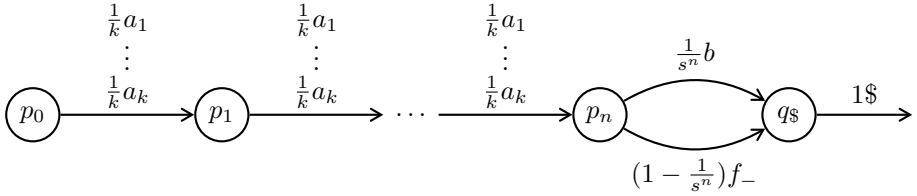
▶ **Lemma 5.** *Given an NFA $\mathcal{A} = (Q, \Sigma, \delta, q^{(1)}, F)$ and a number $n \in \mathbb{N}$ in unary, one can compute in polynomial time an acyclic LMC $\mathcal{M}$ and initial distributions π_1, π_2 and a rational number y such that*

$$d(\pi_1, \pi_2) \;=\; y + \frac{|\Sigma^n \setminus L(\mathcal{A})|}{|\Sigma|^n |Q|^n}\,.$$

Proof. In the following we assume $Q = \{q^{(1)}, \ldots, q^{(s)}\}$ and $\Sigma = \{a_1, \ldots, a_k\}$. Construct the acyclic LMC $\mathcal{M} = (Q', \Sigma \cup \{b, f_+, f_-\}, M)$ such that

$$Q' \;=\; \{p_0, p_1, \ldots, p_n, q_\$\} \;\cup\; \{q_i^{(j)} : 0 \le i \le n,\ 1 \le j \le s\} \;\cup\; \{r_i : 0 \le i \le n\}$$

and b, f_+, f_- are fresh labels. The transitions and end-of-word probabilities originating in the states $p_0, \ldots, p_n, q_\$$ are as follows:



Define $\pi_1 := \delta_{p_0}$. Then for all $w \in \Sigma^n$ we have:

$$\pi_1(wb) \;=\; \frac{1}{k^n} \cdot \frac{1}{s^n} \tag{3}$$

$$\pi_1(wf_-) \;=\; \frac{1}{k^n} \cdot \Big(1 - \frac{1}{s^n}\Big) \tag{4}$$

The transitions originating in the states $q_i^{(j)}, r_i$ are as follows. For each $a \in \Sigma$ and each $i \in \{0, \ldots, n-1\}$ set:

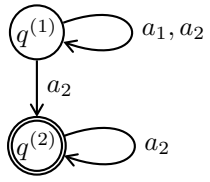
$$M(a)\big(q_i^{(j)}, q_{i+1}^{(j')}\big) \;:=\; \frac{1}{k} \cdot \frac{1}{s} \qquad \forall\, j \in \{1, \ldots, s\} \quad \forall\, q^{(j')} \in \delta(q^{(j)}, a)$$

$$M(a)\big(q_i^{(j)}, r_{i+1}\big) \;:=\; \frac{1}{k} \cdot \Big(1 - \frac{|\delta(q^{(j)}, a)|}{s}\Big) \qquad \forall\, j \in \{1, \ldots, s\}$$

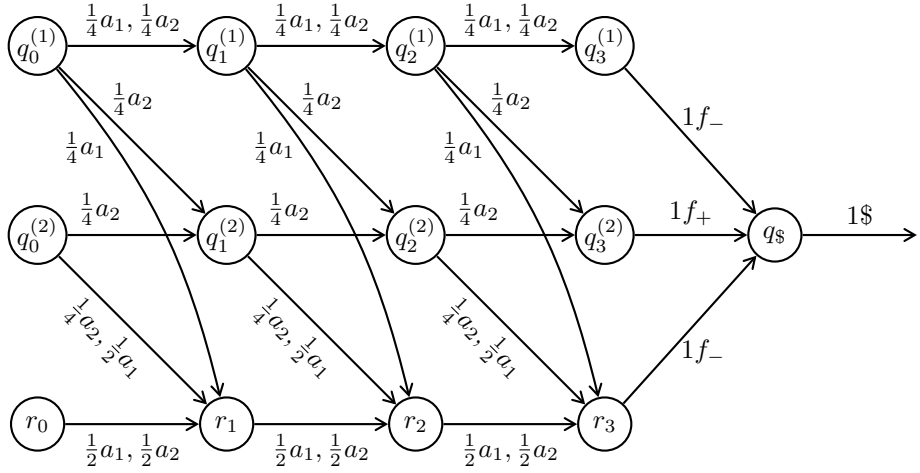
$$M(a)\big(r_i, r_{i+1}\big) \;:=\; \frac{1}{k}$$

Observe that if $i \in \{0, \ldots, n-1\}$ then r_i and all $q_i^{(j)}$ emit each $a \in \Sigma$ with probability $1/k$. For each $q^{(j)} \in F$ set $M(f_+)(q_n^{(j)}, q_\$) := 1$. For each $q^{(j)} \notin F$ set $M(f_-)(q_n^{(j)}, q_\$) := 1$. Finally, set $M(f_-)(r_n, q_\$) := 1$.

▶ **Example 6.** We illustrate this construction with the following NFA $\mathcal{A}$ over $\Sigma = \{a_1, a_2\}$:



For $n = 3$ we obtain the following transitions:



Define $\pi_2 := \delta_{q_0^{(1)}}$. For all $w \in \Sigma^*$ write $\#acc(w)$ for the number of accepting w-labelled runs of the automaton $\mathcal{A}$, i.e., the number of w-labelled paths from $q^{(1)}$ to a state in F. For all $w \in \Sigma^n$ we have:

$$\pi_2(wf_+) \;=\; \frac{1}{k^n} \cdot \frac{\#acc(w)}{s^n} \tag{5}$$

$$\pi_2(wf_-) \;=\; \frac{1}{k^n} \cdot \Big(1 - \frac{\#acc(w)}{s^n}\Big) \tag{6}$$

Define $B := \Sigma^n\{b, f_-\}$. By (3), (4) we have $\pi_1(B) = 1$. One can compute $\pi_2(B)$ in polynomial time by computing the probability of reaching a transition labelled by f_- (the label b is not reachable). Set $y := \pi_1(B) - \pi_2(B)$.

It follows from Proposition 2 that $d(\pi_1, \pi_2) = \pi_1(L) - \pi_2(L)$ holds for

$$L \;:=\; \{w \in (\Sigma \cup \{b, f_+, f_-\})^* : 0 < \pi_1(w) \geq \pi_2(w)\}\,.$$

Observe that $L(\mathcal{A}) = \{w \in \Sigma^n : \#acc(w) \geq 1\}$. Hence it follows with (3), (4), (6):

$$L \;=\; \Sigma^n\{b\} \;\cup\; (\Sigma^n \cap L(\mathcal{A}))\{f_-\}$$

Defining $\overline{L(\mathcal{A})} := \Sigma^n \setminus L(\mathcal{A})$ we can write:

$$L \;=\; B \setminus \big(\overline{L(\mathcal{A})}\{f_-\}\big)$$

Thus we have:

$$\begin{aligned} d(\pi_1, \pi_2) \;&=\; \pi_1\Big(B \setminus \big(\overline{L(\mathcal{A})}\{f_-\}\big)\Big) - \pi_2\Big(B \setminus \big(\overline{L(\mathcal{A})}\{f_-\}\big)\Big) && \text{as argued above} \\ &=\; y + \pi_2\big(\overline{L(\mathcal{A})}\{f_-\}\big) - \pi_1\big(\overline{L(\mathcal{A})}\{f_-\}\big) && \text{definition of } y \end{aligned}$$

Observe that $\overline{L(\mathcal{A})} = \{w \in \Sigma^n : \#acc(w) = 0\}$. Hence we can continue:

$$\begin{aligned} &=\; y + \frac{\big|\overline{L(\mathcal{A})}\big|}{k^n} - \frac{\big|\overline{L(\mathcal{A})}\big|}{k^n} \cdot \Big(1 - \frac{1}{s^n}\Big) && \text{by (6), (4)} \\ &=\; y + \frac{\big|\overline{L(\mathcal{A})}\big|}{k^n s^n} \;=\; y + \frac{|\Sigma^n \setminus L(\mathcal{A})|}{|\Sigma|^n |Q|^n} && \text{definitions} \end{aligned}$$

◀

The PP lower bound from Proposition 4 is tight for acyclic LMCs:

▶ **Theorem 7.** *The non-strict and strict threshold-distance problems are PP-complete for acyclic LMCs.*

▶ **Remark 8.** The works [20, 7] also consider the L_k-distances for integers k:

$$d_k(\pi_1, \pi_2) \;:=\; \sum_{w \in \Sigma^*} |\pi_1(w) - \pi_2(w)|^k$$

For any fixed even k one can compute d_k in polynomial time, see, e.g., [7, Theorem 6]. In contrast, it is NP-hard to compute or even approximate d_k for any odd k [7, Theorems 7 and 10]. Our PP- and #P-hardness results (Proposition 4 and Theorem 9) hold for d_1 (due to Proposition 2) but the reductions do not apply in an obvious way to d_k for any $k \geq 2$. However, the argument in the proof of Theorem 7 for the PP upper bound does generalize to all d_k, see [15].

4 Approximation

As the strict threshold-distance problem is undecidable (Theorem 3), one may ask whether the distance can be approximated. It is not hard to see that the answer is yes. In fact, it was shown in [5, Corollary 8] that the distance can be approximated within an arbitrary additive error even for *infinite-word* LMCs, but no complexity bounds were given. In this section we provide bounds on the complexity of approximating the distance for (finite-word) LMCs.

4.1 Hardness

Lemma 5 implies hardness of approximating the distance:

▶ **Theorem 9.** *Given an LMC and initial distributions π_1, π_2 and an error bound $\varepsilon > 0$ in binary, it is #P-hard to compute a number x with $|d(\pi_1, \pi_2) - x| \leq \varepsilon$, even for acyclic LMCs.*

Proof. Recall that the problem #NFA is #P-complete [14]. Let $\mathcal{A}$ be the given NFA and $n \in \mathbb{N}$. Let $\mathcal{M}, \pi_1, \pi_2, y$ be as in Lemma 5. Approximate $d(\pi_1, \pi_2)$ within $1/(3|\Sigma|^n|Q|^n)$ and call the approximation $\tilde{d}$. It follows from Lemma 5 that $|L(\mathcal{A}) \cap \Sigma^n|$ is the unique integer u with

$$\left| y + \frac{|\Sigma|^n - u}{|\Sigma|^n |Q|^n} - \tilde{d} \right| \;\leq\; \frac{1}{3|\Sigma|^n|Q|^n} \,.$$

Such u can be computed in polynomial time. ◀

Theorem 9 improves the NP-hardness result of [5, Proposition 9]. In fact, PP and #P are substantially harder than NP: By Toda's theorem [25], the polynomial-time hierarchy (PH) is contained in $\mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$. Therefore, any problem in PH can be decided in deterministic polynomial time with the help of an oracle for the threshold-distance problem or for approximating the distance.

4.2 Acyclic LMCs

Towards approximation algorithms, define $W_2 := \{w \in \Sigma^* : \pi_1(w) \geq \pi_2(w)\}$ and $W_1 := \{w \in \Sigma^* : \pi_1(w) < \pi_2(w)\}$. By Proposition 2 we have:

$$d(\pi_1, \pi_2) \;=\; \pi_1(W_2) - \pi_2(W_2) \;=\; 1 - \pi_1(W_1) - \pi_2(W_2) \tag{7}$$

Therefore, to approximate $d(\pi_1, \pi_2)$ it suffices to approximate $\pi_i(W_i)$. A simple sampling scheme leads to the following theorem:

▶ **Theorem 10.** *There is a randomized algorithm, R, that, given an acyclic LMC $\mathcal{M}$ and initial distributions π_1, π_2 and an error bound $\varepsilon > 0$ and an error probability $\delta \in (0,1)$, does the following:*

- *R computes, with probability at least $1-\delta$, a number x with $|d(\pi_1,\pi_2) - x| \leq \varepsilon$;*
- *R runs in time polynomial in $\frac{\log \delta}{\varepsilon}$ and in the encoding size of $\mathcal{M}$ and π_1, π_2.*

Note that $\frac{1}{\varepsilon}$ is not polynomial in the bit size of ε, so combining Theorems 9 and 10 does not imply breakthroughs in computational complexity.

Proof. Let $i \in \{1,2\}$. The length of a longest word w with $\pi_i(w) > 0$ is polynomial in the encoding of the (acyclic) LMC $\mathcal{M}$. Thus, one can sample, in time polynomial in the encoding of $\mathcal{M}, \pi_1, \pi_2$, a word w according to $\Pr_{\pi_i}$; i.e., any w is sampled with probability $\pi_i(w)$. Similarly, one can check in polynomial time whether $w \in W_i$. If m samples are taken, the proportion, say $\hat{p}_i$, of samples w such that $w \in W_i$ is an estimation of $\pi_i(W_i)$. By Hoeffding's inequality, we have $|\hat{p}_i - \pi_i(W_i)| \geq \varepsilon/2$ with probability at most $2e^{-m\varepsilon^2/2}$. Choose $m \geq -\frac{2}{\varepsilon^2} \ln \frac{\delta}{4}$. It follows that $|\hat{p}_i - \pi_i(W_i)| > \varepsilon/2$ with probability at most $\delta/2$. Therefore, by (7), the algorithm that returns $1 - \hat{p}_1 - \hat{p}_2$ has the required properties. ◀

4.3 General LMCs

Finally we aim at an algorithm that approximates the distance within ε, for ε given in binary. By Theorem 9 such an algorithm cannot run in polynomial time unless P = PP. For LMCs that are not necessarily acyclic, words of polynomial length may have only small probability, so sampling approaches need to sample words of exponential length. Thus, a naive extension of the algorithm from Theorem 10 leads to a randomized exponential-time algorithm. We will develop a non-randomized PSPACE algorithm, resulting in the following theorem:

▶ **Theorem 11.** *Given an LMC, and initial distributions π_1, π_2, and an error bound $\varepsilon > 0$ in binary, one can compute in PSPACE a number x with $|d(\pi_1,\pi_2) - x| \leq \varepsilon$.*

The approximation algorithm combines special techniques. The starting point is again the expression for the distance in (7). The following lemma allows the algorithm to neglect words that are longer than exponential:

▶ **Lemma 12.** *Given an LMC, and initial distributions π_1, π_2, and a rational number $\lambda > 0$ in binary, one can compute in polynomial time a number $n \in \mathbb{N}$ in binary such that*

$$\pi_i(\Sigma^{>n}) \;\leq\; \lambda \qquad \text{for both } i \in \{1,2\}\,.$$

For n as in Lemma 12 and both $i \in \{1,2\}$, define $W_i' := W_i \cap \Sigma^{\leq n}$. By Lemma 12 it would suffice to approximate $\pi_i(W_i')$ for both i, as we have by (7):

$$\pi_1(W_1') + \pi_2(W_2') \quad\leq\quad 1 - d(\pi_1,\pi_2) \quad\leq\quad \pi_1(W_1') + \pi_2(W_2') + 2\lambda \tag{8}$$

However, it not obvious if $\pi_i(W_i')$ can be approximated efficiently, as for exponentially long words w it is hard to check if $w \in W_i'$ holds. Indeed, $\pi_i(w)$ may be very small and may have exponential bit size. The main trick of our algorithm will be to approximate $\pi_i(w)$ using floating-point arithmetic with small *relative* error, say $\widetilde{\pi}_i(w) \in [\pi_i(w)(1-\theta), \pi_i(w)(1+\theta)]$ for small $\theta > 0$. This allows us to approximate $\pi_1(W_1') + \pi_2(W_2')$ (crucially, not the two summands individually). Indeed, define approximations for W_1' and W_2' by

$$\widetilde{W_1} \;:=\; \{w \in \Sigma^{\leq n} : \widetilde{\pi}_1(w) < \widetilde{\pi}_2(w)\} \qquad \text{and} \qquad \widetilde{W_2} \;:=\; \{w \in \Sigma^{\leq n} : \widetilde{\pi}_1(w) \geq \widetilde{\pi}_2(w)\}\,.$$

Then we have:

$$\begin{aligned}
\pi_2(w) \;\le\; \pi_1(w) \;<\; \pi_2(w) + \theta\pi_1(w) + \theta\pi_2(w) \qquad & \text{for } w \in \widetilde{W}_1 \cap W_2' \\
\pi_1(w) \;<\; \pi_2(w) \;\le\; \pi_1(w) + \theta\pi_1(w) + \theta\pi_2(w) \qquad & \text{for } w \in \widetilde{W}_2 \cap W_1'
\end{aligned}$$

It follows:

$$\begin{aligned}
\pi_2(\widetilde{W}_1 \cap W_2') \;\le\; \pi_1(\widetilde{W}_1 \cap W_2') \;\le\; \pi_2(\widetilde{W}_1 \cap W_2') + 2\theta \\
\pi_1(\widetilde{W}_2 \cap W_1') \;\le\; \pi_2(\widetilde{W}_2 \cap W_1') \;\le\; \pi_1(\widetilde{W}_2 \cap W_1') + 2\theta
\end{aligned} \tag{9}$$

Hence we have:

$$\begin{aligned}
\pi_1(W_1') + \pi_2(W_2') \;&=\; \pi_1(\widetilde{W}_1 \cap W_1') + \pi_2(\widetilde{W}_1 \cap W_2') + \pi_2(\widetilde{W}_2 \cap W_2') + \pi_1(\widetilde{W}_2 \cap W_1') \\
&\overset{(9)}{\le}\; \pi_1(\widetilde{W}_1) + \pi_2(\widetilde{W}_2) \\
&\overset{(9)}{\le}\; \pi_1(W_1') + \pi_2(W_2') + 4\theta
\end{aligned}$$

By combining this with (8) we obtain:

$$\pi_1(\widetilde{W}_1) + \pi_2(\widetilde{W}_2) - 4\theta \quad\le\quad 1 - d(\pi_1, \pi_2) \quad\le\quad \pi_1(\widetilde{W}_1) + \pi_2(\widetilde{W}_2) + 2\lambda \tag{10}$$

It remains to tie two loose ends:

1. develop a PSPACE method to approximate $\pi_i(w)$ within *relative* error θ for any $\theta > 0$ in binary, where w is an at most exponentially long word (given on a special input tape);
2. based on this method, approximate $\pi_i(\widetilde{W}_i)$ in PSPACE.

For item 1 we use floating-point arithmetic, for item 2 we use Ladner's result [19] on counting in polynomial space.

For $k \in \mathbb{N}$, define $\mathbb{F}_k := \{m \cdot 2^z : z \in \mathbb{Z},\ 0 \le m \le 2^k - 1\}$, the set of *$k$-bit floating-point numbers*. For our purposes, nonnegative floating-point numbers suffice, and there is no need to bound the exponent z, as all occurring exponents will have polynomial bit size. We define rounding as usual: for $x \ge 0$ write $\langle x \rangle_k$ for the number in $\mathbb{F}_k$ that is nearest to x (break ties in an arbitrary but deterministic way). Then there is δ with $\langle x \rangle_k = x \cdot (1 + \delta)$ and $|\delta| < 2^{-k}$, see [13, Theorem 2.2]. A standard analysis of rounding errors in finite-precision arithmetic [13, Chapter 3] yields the following lemma:

▶ **Lemma 13.** *Let π be an initial distribution and $0 < \theta < 1$. Let $k \in \mathbb{N}$ be such that $2^k \ge 2(n+1)|Q|/\theta$. Let $w = a_1 a_2 \cdots a_m \in \Sigma^*$ with $m \le n$. Compute $\widetilde{\pi}(w)$ as*

$$((\cdots((\pi \cdot M(a_1)) \cdot M(a_2)) \cdots) \cdot M(a_m)) \cdot \eta^\top ,$$

where rounding $\langle \cdot \rangle_k$ is applied after each individual (scalar) multiplication and addition. Then $\widetilde{\pi}(w) \in [\pi(w)(1-\theta), \pi(w)(1+\theta)]$.

Proof. For all $i \in \mathbb{N}$ write $\gamma_i := i \cdot 2^{-k}/(1 - i \cdot 2^{-k})$. By [13, Equation (3.11)] there are matrices $\Delta_1, \ldots, \Delta_m$ and a vector $\widetilde{\eta}$ such that

$$\widetilde{\pi}(w) \;=\; \pi \cdot (M(a_1) + \Delta_1) \cdot (M(a_2) + \Delta_2) \cdots (M(a_m) + \Delta_m) \cdot (\eta + \widetilde{\eta})^\top$$

and $|\Delta_i| \le \gamma_{|Q|} M(a_i)$ and $|\widetilde{\eta}| \le \gamma_{|Q|}\eta$, where by $|\Delta_i|$ and $|\widetilde{\eta}|$ we mean the matrix and vector obtained by taking the absolute value componentwise. (In words, the result $\widetilde{\pi}(w)$ of the

floating-point computation is the result of applying an exact computation with slightly perturbed data—a "backward error" result.) It follows:

$$\begin{aligned}
|\widetilde{\pi}(w) - \pi(w)| &\leq \left(-1 + \prod_{j=1}^{m+1} \left(1 + \gamma_{|Q|}\right) \right) \pi(w) && \text{by [13, Lemma 3.8]} \\
&\leq \gamma_{(m+1)\cdot|Q|}\pi(w) && \text{by [13, Lemma 3.3]} \\
&\leq 2(n+1)|Q| \cdot 2^{-k}\pi(w) && \text{as } (n+1)|Q| \cdot 2^{-k} \leq 1/2 \\
&\leq \theta\pi(w)
\end{aligned}$$

◀

The development so far suggests the following approximation approach: Let $\varepsilon > 0$ be the error bound from the input. Let $n \in \mathbb{N}$ be the number from Lemma 12, where λ is such that $2\lambda = \varepsilon/2$. Let $k \in \mathbb{N}$ be the smallest number such that $2^k \geq 2(n+1)|Q|/\theta$, where θ is such that $4\theta = \varepsilon/2$. Observe that k (the bit size of 2^k) is polynomial in the input. Define, for each word w and both i, the approximation $\widetilde{\pi}_i(w)$ as in Lemma 13. This defines also $\widetilde{W}_1, \widetilde{W}_2$. By (10) we have:

$$\pi_1(\widetilde{W}_1) + \pi_2(\widetilde{W}_2) - \frac{\varepsilon}{2} \quad \leq \quad 1 - d(\pi_1, \pi_2) \quad \leq \quad \pi_1(\widetilde{W}_1) + \pi_2(\widetilde{W}_2) + \frac{\varepsilon}{2}$$

Thus we can complete the proof of Theorem 11 by proving the following lemma:

▶ **Lemma 14.** *For both i, one can approximate $\pi_i(\widetilde{W}_i)$ within $\varepsilon/4$ in PSPACE.*

Proof. We discuss only the approximation of $\pi_1(\widetilde{W}_1)$; the case of $\pi_2(\widetilde{W}_2)$ is similar.

Construct a "probabilistic PSPACE Turing machine" $\mathcal{T}$ that samples a random word w according to $\Pr_{\pi_1}$. For that, $\mathcal{T}$ uses probabilistic branching according to the transition probabilities in M. While producing w in this way, but without storing w as a whole, $\mathcal{T}$ computes also the values $\widetilde{\pi}_1(w), \widetilde{\pi}_2(w)$ according to Lemma 13. If and when w gets longer than n then $\mathcal{T}$ rejects. If $\widetilde{\pi}_1(w) < \widetilde{\pi}_2(w)$ then $\mathcal{T}$ accepts; otherwise $\mathcal{T}$ rejects. The probability that $\mathcal{T}$ accepts equals $\pi_1(\widetilde{W}_1)$. This probability can be computed in PSPACE by Ladner's result [19] on counting in polynomial space. To be precise, note that this probability is a fraction p/q of two natural numbers p, q of at most exponential bit size. By Ladner's result one can compute arbitrary bits of p, q in PSPACE. Hence an approximation within $\varepsilon/4$ can also be computed in PSPACE. Technical details about how we apply Ladner's result are provided in [15]. ◀

5 Open Problems

In this paper we have considered the total variation distance between the distributions on finite words that are generated by two LMCs. In a more general version of LMCs, the end-of-word probabilities are zero, so that the LMC generates infinite words. The production of finite words $w \in \Sigma^*$ can be simulated by producing $w\$\$\$\cdots$ where \$ is an end-of-word symbol. It follows that the undecidability and hardness results of this paper apply equally to infinite-word LMCs. In fact, all these results strengthen those from [5], where the total variation distance between infinite-word LMCs is studied. The PSPACE approximation algorithm in this paper (Theorem 11) applies only to finite words, and the author does not know if it can be generalized to infinite-word LMCs. Whether the non-strict threshold-distance problem is decidable is open, both for finite- and for infinite-word LMCs.

Another direction concerns LMCs that are *not hidden*, i.e., where each emitted label identifies the next state; or, slightly more general, *deterministic* LMCs, i.e., where each state

and each emitted label identify the next state. The reduction that shows square-root-sum hardness in [5, Theorem 15] also applies to the threshold-distance problem for deterministic finite-word LMCs, but the author does not know a hardness result for approximating the distance between deterministic LMCs.

References

1 P. Ailliot, C. Thompson, and P. Thomson. Space-time modelling of precipitation by using a hidden Markov model and censored Gaussian distributions. *Journal of the Royal Statistical Society*, 58(3):405–426, 2009.

2 M. Alexandersson, S. Cawley, and L. Pachter. SLAM: Cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Research*, 13:469–502, 2003.

3 D. Chen, F. van Breugel, and J. Worrell. On the complexity of computing probabilistic bisimilarity. In *Proceedings of FoSSaCS*, volume 7213 of *LNCS*, pages 437–451. Springer, 2012.

4 F.-S. Chen, C.-M. Fu, and C.-L. Huang. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision Computing*, 21(8):745–758, 2003.

5 T. Chen and S. Kiefer. On the total variation distance of labelled Markov chains. In *Proceedings of CSL-LICS*, pages 33:1–33:10, 2014.

6 G.A. Churchill. Stochastic models for heterogeneous DNA sequences. *Bulletin of Mathematical Biology*, 51(1):79–94, 1989.

7 C. Cortes, M. Mohri, and A. Rastogi. L_p distance and equivalence of probabilistic automata. *International Journal of Foundations of Computer Science*, 18(04):761–779, 2007.

8 M.S. Crouse, R.D. Nowak, and R.G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 46(4):886–902, April 1998.

9 C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk. A Storm is coming: A modern probabilistic model checker. In *Proceedings of Computer Aided Verification (CAV)*, pages 592–600. Springer, 2017.

10 J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

11 R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

12 S.R. Eddy. What is a hidden Markov model? *Nature Biotechnology*, 22(10):1315–1316, October 2004.

13 N. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, second edition, 2002.

14 S. Kannan, Z. Sweedyk, and S. Mahaney. Counting and random generation of strings in regular languages. In *Proceedings of SODA*, pages 551–557, 1995.

15 S. Kiefer. On computing the total variation distance of hidden Markov models. Technical report, arxiv.org, 2018. Available at `https://arxiv.org/abs/1804.06170`.

16 S. Kiefer, A.S. Murawski, J. Ouaknine, B. Wachter, and J. Worrell. Language equivalence for probabilistic automata. In *Proceedings of Computer Aided Verification (CAV)*, volume 6806 of *LNCS*, pages 526–540. Springer, 2011.

17 A. Krogh, B. Larsson, G. von Heijne, and E.L.L. Sonnhammer. Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580, 2001.

18 M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of Computer Aided Verification (CAV)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

19 R. E. Ladner. Polynomial space counting problems. *SIAM Journal on Computing*, 18(6):1087–1097, 1989.

20 R.B. Lyngsø and C.N.S. Pedersen. The consensus string problem and the complexity of comparing hidden Markov models. *J. Comput. Syst. Sci.*, 65(3):545–569, 2002.

21 M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

22 A. Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

23 L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

24 M.-P. Schützenberger. On the definition of a family of automata. *Inf. and Control*, 4:245–270, 1961.

25 S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal of Computing*, 20(5):865–877, 1991.