International Journal of Foundations of Computer Science

On the Computation of the Relative Entropy of Probabilistic Automata

Corinna Cortes

Google Research,
76 Ninth Avenue, New York, NY 10011.
corinna@google.com

Mehryar Mohri

Courant Institute of Mathematical Sciences and Google Research,
251 Mercer Street, New York, NY 10012.
mohri@cims.nyu.edu

Ashish Rastogi

Courant Institute of Mathematical Sciences,
251 Mercer Street, New York, NY 10012.
rastogi@cs.nyu.edu

Michael Riley

Google Research,
76 Ninth Avenue, New York, NY 10011.
riley@google.com

We present an exhaustive analysis of the problem of computing the relative entropy of two probabilistic automata. We show that the problem of computing the relative entropy of unambiguous probabilistic automata can be formulated as a shortest-distance problem over an appropriate semiring, give efficient exact and approximate algorithms for its computation in that case, and report the results of experiments demonstrating the practicality of our algorithms for very large weighted automata. We also prove that the computation of the relative entropy of arbitrary probabilistic automata is PSPACE-complete.

The relative entropy is used in a variety of machine learning algorithms and applications to measure the discrepancy of two distributions. We examine the use of the symmetrized relative entropy in machine learning algorithms and show that, contrarily to what is suggested by a number of publications in that domain, the symmetrized relative entropy is neither positive definite symmetric nor negative definite symmetric, which limits its use and application in kernel methods. In particular, the convergence of training for learning algorithms is not guaranteed when the symmetrized relative entropy is used directly as a kernel, or as the operand of an exponential as in the case of Gaussian Kernels.

Finally, we show that our algorithm for the computation of the entropy of an unambiguous probabilistic automaton can be generalized to the computation of the norm of an unambiguous probabilistic automaton by using a monoid morphism. In particular, this yields efficient algorithms for the computation of the L_p-norm of a probabilistic automaton.

1. Introduction

The problem of comparing two distributions arises in a variety of applications. A specific instance of that problem is that of comparing distributions given by probabilistic automata. Probabilistic automata are used extensively in text and speech processing to model different aspects of language such as morphology, phonology, or syntax [Mohri, 1997, Mohri et al., 1996] or in other applications such as computational biology [Durbin et al., 1998] and image processing [Culik II and Kari, 1997].

The output of a large-vocabulary speech recognition system or that of a complex information extraction system is often represented as a probabilistic automaton compactly representing a large set of alternative sequences [Mohri et al., 2002]. Natural language sequences such as documents or biological sequences can also be modeled by probabilistic automata [Krogh et al., 1994]. The computation of the distance or discrepancy between probabilistic automata can thus be used to cluster the outputs of speech recognition or information extraction systems, documents, biological sequences, or other objects modeled in a similar way.

The problem of efficiently computing the distance between two distributions represented by weighted automata arises in many other machine learning problems. When a weighted automaton is obtained as a result of training on a large data set, the quality of the learning algorithm can be measured by computing the distance between the automaton inferred and that of the target automaton. Similarly, in many on-line learning algorithms and grammar inference applications, the convergence of an iterative algorithm relies on the magnitude of the distance between two consecutive weighted automata.

This motivates the design of efficient algorithms for the computation of the distance or discrepancy between probabilistic automata.[a] There are many standard distances or divergences commonly used to compare distributions, including the relative entropy or Kullback-Leibler divergence, the L_p distance, the Hellinger distance, the Jensen-Shannon distance, the χ^2-distance, and the Triangle distance between two distributions q_1 and q_2 defined over a discrete set $\mathcal{X}$ [Topsøe, 2000, Csiszar and Korner, 1997].

In a companion paper, we give an exhaustive study of the problem of computing the L_p distance of two probabilistic automata and other similar distances such as the Hellinger distance [Cortes et al., 2007]. In particular, we give efficient exact and approximate algorithms for computing these distances for p even and prove the problem to be NP-hard for all odd values of p, thereby completing previously known hardness results. We also show the hardness of approximating the L_p distance of two probabilistic automata for odd values of p.

This paper deals with the problem of computing the relative entropy of two

[a] A related problem is that of testing the equivalence of two arbitrary probabilistic automata A_1 and A_2. In [Cortes et al., 2006, 2007], we give an efficient algorithm for this problem whose time complexity is $O(|\Sigma|\,(|A_1| + |A_2|)^3)$, where Σ is the alphabet.

probabilistic automata. The relative entropy, or Kullback-Leibler divergence, is one of the most commonly used measures of the discrepancy of two distributions p and q [Cover and Thomas, 1991]. It is an asymmetric difference that admits the following information-theoretical interpretation: it measures the number of additional bits needed to encode distribution p when using an optimal code for q in place of an optimal code for p.

One approximate solution for the computation of the relative entropy would consist of sampling sequences from the distributions represented by each of the automata and of using those to compute the KL-divergence by simply summing their contributions. But, sample sizes guaranteeing a small approximation error could be very large, which would significantly increase the computation, while still providing only an approximate solution.

We present an exhaustive analysis of the problem of computing the relative entropy of two probabilistic automata. We show that the problem of computing the relative entropy of unambiguous probabilistic automata can be formulated as a shortest-distance problem over an appropriate semiring, give efficient exact and approximate algorithms for its computation in that case, and report the results of experiments demonstrating the practicality of our algorithms for very large weighted automata. We also prove that the computation of the relative entropy of arbitrary probabilistic automata is PSPACE-complete.

A procedure for the approximate computation of the relative entropy was given by Carrasco [1997]. The procedure applies to deterministic weighted automata and cannot be generalized to the case of unambiguous weighted automata because of the specific sum decomposition it is based on (the partitioning assumed in [Carrasco, 1997] [Eqs. 15 and 16, page 6] does not hold for unambiguous automata). Our algorithms apply to the larger class of unambiguous weighted automata. For some unambiguous weighted automata, the size of any equivalent deterministic weighted automaton is exponentially larger. Since the size of the machine directly affects the complexity of the computation, it is important to be able to compute the entropy directly from the unambiguous automaton. We give the first *exact* algorithms for the computation of the relative entropy. We also describe approximate algorithms that are conceptually simpler than the procedure of Carrasco [1997] and have a better time and space complexity.

The relative entropy is used in a variety of machine learning algorithms and applications to measure the discrepancy of two distributions. We examine the use of the symmetrized relative entropy in machine learning algorithms and show that, contrarily to what is suggested by a number of publications (e.g., [Mandel et al., 2006]), the symmetrized relative entropy is neither positive definite symmetric nor negative definite symmetric, which limits its use and application in kernel methods. In particular, the convergence of training for learning algorithms is not guaranteed when the symmetrized relative entropy is used directly as a kernel, or as the operand of an exponential as in the case of Gaussian Kernels [Schölkopf and Smola, 2002].

Finally, we show that our algorithm for the computation of the entropy of an

unambiguous probabilistic automaton can be generalized to the computation of the norm of an unambiguous probabilistic automaton by using a monoid morphism [Cortes et al., 2006]. In particular, this yields efficient algorithms for the computation of the L_p-norm of a probabilistic automaton.

The paper is organized as follows. Section 2 introduces the preliminary semiring and automata definitions used in the remaining of the paper. Section 3 recalls the definition of the relative entropy of two probabilistic automata and introduces a semiring, the *entropy semiring*, which helps formulate the computation of the relative entropy of unambiguous probabilistic automata as a shortest-distance problem. Section 4 describes both an exact and a fast approximate algorithm for the computation of the relative entropy of unambiguous probabilistic automata. It also provides a detailed analysis of these algorithms and reports the results of experiments with large weighted automata. The case of arbitrary probabilistic automata is treated in Section 5 where the problem is proven to be PSPACE-complete. Section 6 proves several negative results for the use of the symmetrized relative entropy in kernel methods. It proves that the symmetrized relative entropy is neither positive definite nor negative definite. Finally, Section 7 extends our algorithm for the computation of the entropy of a probabilistic automaton to the computation of other norms defined via a monoid morphism.

2. Preliminaries

2.1. *Semirings and Weighted Automata*

Weighted automata are automata in which each transition carries some weight in addition to the usual alphabet symbol [Eilenberg, 1974–1976, Salomaa and Soittola, 1978, Berstel and Reutenauer, 1988]. For various operations to be well-defined, the weight set must have the algebraic structure of a semiring [Kuich and Salomaa, 1986]. A semiring is a ring that may lack negation.

Definition 1. *A* semiring *is a system* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ *such that:*

- $(\mathbb{K}, \oplus, \overline{0})$ *is a commutative monoid with* $\overline{0}$ *as the identity element for* $\oplus$*,*
- $(\mathbb{K}, \otimes, \overline{1})$ *is a monoid with* $\overline{1}$ *as the identity element for* $\otimes$*,*
- $\otimes$ *distributes over* $\oplus$*: for all* a, b, c *in* $\mathbb{K}$*,*

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \quad \text{and} \quad c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b).$$

- $\overline{0}$ *is an annihilator for* $\otimes$*:* $\forall a \in \mathbb{K}, a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$.

Some familiar semirings are the Boolean semiring $(\{0,1\}, \vee, \wedge, 0, 1)$ or the tropical semiring $(\mathbb{R}_+ \cup \{\infty\}, \min, +, \infty, 0)$ related to classical shortest-paths problems and algorithms. A semiring is idempotent if for all $a \in \mathbb{K}$, $a \oplus a = a$. It is *commutative* when $\otimes$ is commutative.

Definition 2. *A* weighted automaton $A = (\Sigma, Q, I, F, E, \lambda, \rho)$ *over a semiring* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ *is a 7-tuple where:*

- Σ *is the finite alphabet of the automaton,*
- Q *is a finite set of states,*
- $I \subseteq Q$ *the set of initial states,*
- $F \subseteq Q$ *the set of final states,*
- $E \subseteq Q \times \Sigma \cup \{\epsilon\} \times \mathbb{K} \times Q$ *a finite set of transitions,*
- $\lambda : I \to \mathbb{K}$ *the initial weight function mapping* I *to* $\mathbb{K}$*, and*
- $\rho : F \to \mathbb{K}$ *the final weight function mapping* F *to* $\mathbb{K}$*.*

The weighted automata considered in this paper are assumed not to contain ϵ-transitions. A pre-processing ϵ-removal algorithm can be used to remove such transitions for the automata considered here [Mohri, 2002a]. In the absence of ϵ-cycles, the complexity of that algorithm is in $O(|Q|^2 + |Q||E|)$ [Mohri, 2002a].

We denote by $|A| = |E| + |Q|$ the size of an automaton $A = (\Sigma, Q, I, F, E, \lambda, \rho)$, that is the sum of the number of states and transitions of A. Given a transition $e \in E$, we denote by $i[e]$ its input label, $p[e]$ its origin or previous state and $n[e]$ its destination state or next state, $w[e]$ its weight (weighted automata case). Given a state $q \in Q$, we denote by $E[q]$ the set of transitions leaving q.

A *path* $\pi = e_1 \cdots e_k$ in A is an element of E^* with consecutive transitions: $n[e_{i-1}] = p[e_i]$, $i = 2, \ldots, k$. We extend n and p to paths by setting: $n[\pi] = n[e_k]$ and $p[\pi] = p[e_1]$. A *cycle* is a path with the same origin and destination states. We denote by $P(q, q')$ the set of paths from q to q' and by $P(q, x, q')$ the set of paths from q to q' with input label $x \in \Sigma^*$. The labeling functions i and the weight function w can also be extended to paths by defining the label of a path as the concatenation of the labels of its constituent transitions, and the weight of a path as the $\otimes$-product of the weights of its constituent transitions: $i[\pi] = i[e_1] \cdots i[e_k]$, $w[\pi] = w[e_1] \otimes \cdots \otimes w[e_k]$.

The output weight associated by an automaton A to an input string $x \in \Sigma^*$ is defined by:

$$[\![A]\!](x) = \bigoplus_{\pi \in P(I,x,F)} \lambda[p[\pi]] \otimes w[\pi] \otimes \rho[n[\pi]]. \tag{1}$$

The language denoted by A is denoted by $L(A)$ and defined by: $L(A) = \{x : P(I, x, F) \neq \emptyset\}$.

A state of an automaton A is *accessible* if it can be reached from an initial state. It is said to be *co-accessible* if it lies on a path reaching a final state. An automaton is said to be *trim* if all of its states are both accessible and co-accessible.

2.2. *Deterministic and Unambiguous Weighted automata*

A weighted automaton A is said to be *deterministic* or *subsequential* if it has a deterministic input, that is if it has a unique initial state and if no two transitions leaving the same state share the same input label. A weighted automaton is said to be *unambiguous* if for any $x \in \Sigma^*$ it admits at most one accepting path labeled

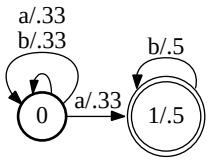

Fig. 1. An unambiguous weighted finite automaton that cannot be determinized. 0 is the initial state and 1 the final state. The automaton accepts the set of strings $(a^*b^*)^*ab^*$.

with x. Thus, the class of unambiguous weighted automata includes *deterministic* weighted automata.

Fig. 1 shows an unambiguous weighted automaton that does not admit an equivalent deterministic weighted automaton. Previous work on the computation of the relative entropy [Carrasco, 1997] was limited to deterministic finite automata. We present the first algorithms for the computation of the relative entropy of unambiguous weighted automata.

2.3. *Shortest-Distances*

Let $s[A]$ denote the $\oplus$-sum of the weights of all successful paths of A when it is defined and in $\mathbb{K}$. $s[A]$ can be viewed as the *shortest-distance* from the initial states to the final states. When the sum of the weights of all paths from any state p to any state q is well-defined and in $\mathbb{K}$, we can define the *shortest distance* from $p \in Q$ to $q \in Q$ as:

$$d[p, q] = \bigoplus_{\pi \in P(p,q)} w[\pi], \tag{2}$$

where the summation is defined to be $\overline{0}$ when $P(p, q) = \emptyset$. When $\oplus$ is replaced by min and $\otimes$ by $+$, this definition coincides with the classical definition of shortest-distance in the tropical semiring. This justifies the terminology used.

2.4. *Probabilistic Automata*

Definition 3. *A weighted automaton A defined over the probability semiring $(\mathbb{R}_+, +, \times, 0, 1)$ is said to be* probabilistic *if for any state $q \in Q$, $\bigoplus_{\pi \in P(q,q)} w[\pi]$, the sum of the weights of all cycles at q, is well-defined and in $\mathbb{R}_+$ and*

$$\sum_{x \in \Sigma^*} [\![A]\!](x) = 1. \tag{3}$$

A probabilistic automaton A is said to be stochastic *if at each state the weights of the outgoing transitions and the final weight sum to one.*

Note that our definition of probabilistic automata differs from that of Rabin [1963] and Paz [1971]. *Probabilistic automata* as defined by these authors are

weighted automata over $(\mathbb{R}_+, +, \times, 0, 1)$ such that at any state q and for any label $a \in \Sigma$, the weights of the outgoing transitions of q labeled with a sum to one. More generally, with that definition, the weights of the paths leaving state q and labeled with $x \in \Sigma^*$ sums to one. Such automata define a conditional probability distribution $\Pr[q' \mid q, x]$ over all states q' that can be reached from q by reading x.

Instead, with our definition, probabilistic automata represent distributions over Σ^*, $\Pr[x], x \in \Sigma^*$. These are the natural distributions that arise in many applications. They are inferred from large data sets using statistical learning techniques. We are interested in computing the relative entropy of two such distributions over strings.

2.5. *Intersection of Weighted Automata*

Let A_1 and A_2 be two weighted automata over the same semiring, with $A_i = (\Sigma, Q_i, I_i, F_i, E_i, \lambda_i, \rho_i)$ for $i = 1, 2$. The intersection A of A_1 and A_2 is denoted by $A = A_1 \cap A_2$. It is a weighted automaton accepting the language $L(A_1) \cap L(A_2)$ and defined by the tuple $A = (\Sigma, Q_1 \times Q_2, I_1 \times I_2, F_1 \times F_2, E, (\lambda_1, \lambda_2), (\rho_1, \rho_2))$, where the transitions E are defined according to the following rule:

$$(q_1, a, w_1, q_2) \in E_1 \text{ and } (q_1', a, w_1', q_2') \in E_2 \Rightarrow ((q_1, q_1'), a, (w_1 \otimes w_1'), (q_2, q_2')) \in E.$$

There exists a general algorithm for the computation of the intersection over an arbitrary semiring, even in the presence of ϵ-transitions [Mohri et al., 1996]. The time complexity of the algorithm is quadratic $O(|A_1||A_2|)$ since in the worst case the outgoing transitions of each state of A_1 match all those of each state of A_2.

3. Relative Entropy

The problem that we are interested in is that of computing $D(A\|B)$, the relative entropy of two unambiguous probabilistic automata A and B.

3.1. *Definition*

The *entropy* $H(p)$ of a probability distribution p defined over a discrete set $\mathcal{X}$ is defined as [Cover and Thomas, 1991]:

$$H(p) = -\sum_{x \in \mathcal{X}} p(x) \log p(x), \tag{5}$$

where by convention $0 \log 0 = 0$. The *relative entropy*, or *Kullback-Leibler divergence* of two probability distributions defined over a discrete set $\mathcal{X}$ is defined as:

$$D(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathrm{E}_p[\log \frac{p(X)}{q(X)}], \tag{6}$$

where we use the standard conventions: $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$. It is straightforward to show, using Jensen's inequality, that the relative entropy is non-negative

and that $D(p\|q) = 0$ if and only if $p = q$. Note that the relative entropy does not define a metric since it is not symmetric and does not satisfy the triangle inequality.

These definitions naturally apply to probabilistic automata since they define distributions over strings. The relative entropy of A and B can be written as the sum of two terms:[b]

$$D(A\|B) = \sum_x [\![A]\!](x) \log [\![A]\!](x) - \sum_x [\![A]\!](x) \log [\![B]\!](x). \tag{7}$$

3.2. *Entropy Semiring*

This section introduces a semiring that will be later used to formulate the problem of computing the relative entropy of two unambiguous automata as a single-source shortest-distance problem.

Let $\mathbb{K}$ denote $(\mathbb{R} \cup \{+\infty, -\infty\}) \times (\mathbb{R} \cup \{+\infty, -\infty\})$. For pairs (x_1, y_1) and (x_2, y_2) in $\mathbb{K}$, define the following :

$$(x_1, y_1) \oplus (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \tag{8}$$

$$(x_1, y_1) \otimes (x_2, y_2) = (x_1 x_2, x_1 y_2 + x_2 y_1) \tag{9}$$

Lemma 4. *The system* $(\mathbb{K}, \oplus, \otimes, (0,0), (1,0))$ *defines a commutative semiring.*

Proof. It is known that $(\mathbb{K}, \oplus, (0,0))$ is a commutative monoid with $(0,0)$ as the identity element for $\oplus$. Furthermore, it is clear that $(\mathbb{K}, \otimes, (1,0))$ is a commutative monoid with $(1,0)$ as the identity element for $\otimes$. Also, $(0,0)$ is an annihilator for $\otimes$. Thus, all that remains to be shown is that $\otimes$ distributes over $\oplus$. Since both operations are commutative, we need to verify that for all $z_1, z_2, z_3 \in \mathbb{K}$,

$$(z_1 \oplus z_2) \otimes z_3 = (z_1 \otimes z_3) \oplus (z_2 \otimes z_3) \tag{10}$$

Let $z_i = (x_i, y_i)$ for $i = 1, 2, 3$. We verify each of these properties one-by-one. First consider $(z_1 \oplus z_2) \otimes z_3$. We have

$$\begin{aligned}
(z_1 \oplus z_2) \otimes z_3 &= ((x_1, y_1) \oplus (x_2, y_2)) \otimes (x_3, y_3) \\
&= (x_1 + x_2, y_1 + y_2) \otimes (x_3, y_3) \\
&= ((x_1 + x_2)x_3, (x_1 + x_2)y_3 + x_3(y_1 + y_2)) \\
&= (x_1 x_3, x_1 y_3 + x_3 y_1) \oplus (x_2 x_3, x_2 y_3 + x_3 y_2) \\
&= ((x_1, y_1) \otimes (x_3, y_3)) \oplus ((x_2, y_2) \otimes (x_3, y_3)) \\
&= (z_1 \otimes z_3) \oplus (z_2 \otimes z_3),
\end{aligned}$$

which ends the proof of the lemma. □

We call the semiring just defined the *entropy semiring* due to its relevance in the computation of the entropy and the relative entropy. This semiring arises in other

[b]The first term is simply $-H(A)$, where $H(A)$ is the entropy of A.

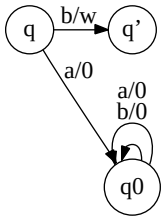

Fig. 2. Illustration of the completion operation.

contexts and can be defined in terms of an S-module [Bloom and Ésik, 1991, Eisner, 2001].

4. Relative Entropy of Unambiguous Probabilistic Automata

This section describes two algorithms for computing the relative entropy of two unambiguous probabilistic automata using a single-source shortest distance over the entropy semiring: an exact algorithm, and a more efficient and practical approximate algorithm. Clearly, these algorithms can also be used to compute the entropy of a single unambiguous probabilistic automaton.

4.1. *Semiring Formulation*

The unambiguous weighted automata A and B are not necessarily *complete*: at some states, there may be no outgoing transition labeled with a given element of the alphabet $a \in \Sigma$. We can however make them complete in a way similar to the standard construction in the unweighted case. We introduce a new state q_0 with final weight 0, add self-loops with weight 0 at that state labeled with all elements of the alphabet, and for any $a \in \Sigma$ and $q \in Q$, add a transition from state q to q_0 labeled with a with weight 0 when q does not have an outgoing transition labeled with a (see Figure 2). This construction leads to a complete and unambiguous weighted automaton equivalent to the original one since the transitions added have all weight 0. The completion operation is only applied to handle the boundary case when there exists a string $x \in \Sigma^*$ such that $[\![B]\!](x) = 0$ and $[\![A]\!](x) \neq 0$. In this case, the completion operation ensures that the future computation of the relative entropy would correctly lead to ∞. Note that the completion operation can be done on-demand. States and transitions can be created only when necessary for the application of other operations. We can thus assume that A and B are unambiguous and complete. At the cost of introducing a super-initial and a super-final state, we can also assume in the following, without loss of generality, that the initial weight λ and the final weights $\rho(q)$ are all equal to 1 in A and B.

Let $\log A$ denote the weighted automaton derived from A by replacing each weight $w \in \mathbb{R}_+$ by $\log w$ and let $\Phi_1(A)$ ($\Phi_2(A)$) denote the weighted automaton over

the entropy semiring derived from A by replacing each weight w by the pair $(w, 0)$ (resp. $(1, w)$). The construction of $\log A$, $\Phi_1(A)$, or $\Phi_2(A)$ from A is straightforward and can be done in linear time.

Proposition 5. *The relative entropy of A and B satisfies the following identity in the entropy semiring:*

$$(0, D(A\|B)) = s[\Phi_1(A) \cap \Phi_2(\log A)] - s[\Phi_1(A) \cap \Phi_2(\log B)]. \tag{11}$$

Thus, the relative entropy is expressed in terms of single-source shortest-distance computations over the entropy semiring.

Proof. Since A is unambiguous and complete, both $\Phi_1(A)$ and $\Phi_2(\log A)$ are also unambiguous and complete. Thus, for a given string x, there is at most one accepting path in $\Phi_1(A)$ or $\Phi_2(\log A)$ labeled with x. Then, by definition of intersection, the weight associated by $\Phi_1(A) \cap \Phi_2(\log A)$ to a string x is

$$([\![A]\!](x), 0) \otimes (1, \log[\![A]\!](x)) = ([\![A]\!](x), [\![A]\!](x) \log[\![A]\!](x)). \tag{12}$$

Thus, the shortest-distance from the initial states to the final states in $\Phi_1(A) \cap \Phi_2(\log A)$ is

$$\begin{aligned} s[\Phi_1(A) \cap \Phi_2(\log A)] &= \bigoplus_x ([\![A]\!](x), [\![A]\!](x) \log[\![A]\!](x)) && (13)\\ &= (\sum_x [\![A]\!](x), \sum_x [\![A]\!](x) \log[\![A]\!](x)) && (14)\\ &= (1, \sum_x [\![A]\!](x) \log[\![A]\!](x)). && (15) \end{aligned}$$

Similarly, we can show that[c]

$$s[\Phi_1(A) \cap \Phi_2(\log B)] = (1, \sum_x [\![A]\!](x) \log[\![B]\!](x)). \tag{16}$$

The statement of the proposition follows directly from the identities 15 and 16 and Equation 7. □

Thus, the computation of the relative entropy is reduced to two single-source shortest-distance computations over the entropy semiring. The next section discusses two general algorithms for computing these distances. Since the first term simply corresponds to the entropy of a single unambiguous probabilistic automaton, our results clearly also apply to the computation of the entropy.

[c]Given a string $x = x_1x_2$ whose respective transitions have weights u_1 and u_2 in A and v_1 and v_2 in B, the weight in $\Phi_1(A) \cap \Phi_2(\log B)$ becomes $(u_1, u_1 \log v_1) \otimes (u_2, u_2 \log v_2) = (u_1u_2, u_1u_2 \log(v_1v_2))$, that is $(\|A\|(x_1x_2), \|A\|(x_1x_2) \log \|B\|(x_1x_2))$.

4.2. *Exact Algorithm*

A generalization of the classical Floyd-Warshall algorithm can be used to compute all-pairs shortest distances $d[p,q]$ $(p,q \in Q)$ over a *closed semiring* not necessarily idempotent [Mohri, 1998, 2002b]. This algorithm can thus also be used to compute $s[A]$ for a weighted automaton A over a non-idempotent semiring, which is needed for our purpose.

In what follows, we assume a definition of closed semirings [Lehmann, 1977] that is more general than the classical one used by Cormen *et al.* [Cormen et al., 1992] in that it does not assume idempotence. This is because idempotence is not necessary for the proof of the correctness of the generic all-pairs shortest-distance algorithms of Floyd-Warshall and Gauss-Jordan [Mohri, 1998, 2002b]. More generally, given a graph or automaton A, we introduce the following definition.

Definition 6. *A semiring is* closed for A *if the infinite sum (closure) is defined for any cycle weight c of A and if associativity, commutativity, and distributivity apply to countable sums of cycle weights.*

Clearly, the generic Floyd-Warshall algorithm can also be applied to any automaton A for which the semiring considered is closed. The following lemma shows that the entropy semiring has the desired property.

Lemma 7. *Let A be a weighted automaton over the entropy semiring such that for any cycle weight $w = (x, y)$, x less than one ($0 \leq x < 1$). Then, the entropy semiring is closed for A.*

Proof. For any $(x, y) \in \mathbb{K}$ and $k \geq 0$, define R_k as:

$$R_k = \overbrace{(x,y) \otimes \ldots \otimes (x,y)}^{k \text{ times}}, \tag{17}$$

with $R_0 = (1, 0)$. It is straightforward to show by induction that $R_k = (x^k, kyx^{k-1}) = (x^k, y\frac{d(x^k)}{dx})$. For $N \geq 0$, define S_N by:

$$S_N = \bigoplus_{i=0}^{N} R_i = \left(\frac{1 - x^{N+1}}{1-x}, y \cdot \left[\frac{1-x^N}{(1-x)^2} - \frac{Nx^N}{1-x} \right] \right). \tag{18}$$

Thus, for $0 \leq x < 1$, the closure of (x, y) is well-defined and in $\mathbb{K}$:[d]

$$(x,y)^* = \lim_{N \to \infty} S_N = \left(\frac{1}{1-x}, \frac{y}{(1-x)^2} \right) = \left(\frac{1}{1-x}, y\frac{d}{dx}(\frac{1}{1-x}) \right). \tag{19}$$

The associativity, commutativity, and distributivity properties follow the associativity, commutativity, and distributivity of the sums S_N with other elements of the entropy semiring and the corresponding properties of their pointwise limits. □

[d]The right-hand side can also be written as: $(x^*, y(x^*)^2)$, if we denote by $x^* = \sum_{n=0}^{\infty} x^n$.

Let A be a probabilistic automaton, then the weight u of a cycle must verify $0 \leq u < 1$, otherwise the automaton is not closed. The weight of a cycle of $\Phi_1(A) \cap \Phi_2(\log A)$ is of the form $(u, u \log u)$ (see Equation 12), where u is the weight of the cycle of A, and similarly, the weight of a cycle of $\Phi_1(A) \cap \Phi_2(\log B)$ is of the form $(u, u \log v)$, where v is the weight of a matching cycle in B.

Thus, the entropy semiring is closed both for $\Phi_1(A) \cap \Phi_2(\log B)$ and $\Phi_1(A) \cap \Phi_2(\log A)$ and the generic Floyd-Warshall algorithm can be applied to compute the shortest-distances $s[\Phi_1(A) \cap \Phi_2(\log B)]$ and $s[\Phi_1(A) \cap \Phi_2(\log A)]$.

The generic Floyd-Warshall admits an in-place implementation [Mohri, 1998]; the following gives the corresponding pseudocode.

```
1 for i ← 1 to |Q|
2     do for j ← 1 to |Q|
3             do d[i, j] ← ⨁_{e∈P(i,j)} w[e]
4 for k ← 1 to |Q|
5     do for i ← 1 to |Q|
6             do for j ← 1 to |Q|
7                     do d[i, j] ← d[i, j] ⊕ (d[i, k] ⊗ d[k, k]* ⊗ d[k, j])
8 return d
```

The $\oplus$- and $\otimes$-operations of the entropy semiring can be performed in constant time. For (x, y) with $0 \leq x < 1$, the closure $(x, y)^* = (\frac{1}{1-x}, \frac{y}{(1-x)^2})$ can also be computed in constant time. Thus, the running time complexity of the algorithm is $\Theta(|E| + |Q|^3)$ and its space complexity is $\Omega(|Q|^2)$ when applied to a weighted automaton $A = (Q, I, F, \Sigma, \delta, \sigma, \lambda, \rho)$ over the tropical semiring.

The intersection $\Phi_1(A) \cap \Phi_2(\log A)$ can be computed in linear time $O(|A|)$ but the worst cost computation of $\Phi_1(A) \cap \Phi_2(\log B)$ is quadratic, $O(|A||B|)$. The total time complexity of the computation of the relative entropy is thus in $\Theta(|A \cap B|^3)$. Its space complexity is in $\Theta(|A \cap B|^2)$.

This provides an exact algorithm for the computation of the relative entropy. The cubic time complexity of the algorithm with respect to the size of the intersection automaton makes it rather slow for large automata.

Its quadratic lower bound complexity with respect to the size of the intersection machine makes it prohibitive for use in many applications. In text and speech processing applications, a weighted automaton may have several hundred million states and transitions. Even, if A has only about 100,000 states and $A \cap B$ has about the same number of states, the algorithm requires maintaining a matrix d with 10 billion entries.

The next section presents an algorithm that exploits the sparseness of the graph and does not impose these space requirements.

4.3. *Approximate Algorithm*

A generic single-source shortest-distance algorithm was presented for directed graphs defined over a *k-closed* semiring in [Mohri, 2002b]. The algorithm can be viewed as a generalization to these semirings of classical shortest-paths algorithms. This generalization is not trivial and does not require the semiring to be idempotent. The algorithm is also generic in the sense that it works with any queue discipline.

Definition 8. *Let $k \geq 0$ be an integer. A semiring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ is* k-closed *if:*

$$\forall a \in \mathbb{K}, \bigoplus_{n=0}^{k+1} a^n = \bigoplus_{n=0}^{k} a^n. \tag{20}$$

More generally, we will say that $\mathbb{K}$ is *k-closed for a graph G or automaton A*, if Equation 20 holds for all cycle weights $a \in \mathbb{K}$.

By definition, the entropy semiring is k-closed for any value of k for any acyclic automaton A and thus the generic single-source shortest distance can be used to compute the relative entropy exactly in such cases. But, in general, the entropy semiring is not k-closed for a non-acyclic automaton A since by definition of S_N,

$$\forall k > 0, S_{k+1} - S_k = R_{k+1} = (x^{k+1}, (k+1)yx^k). \tag{21}$$

But, given a weighted automaton A over the entropy semiring such that all cycle weights $w = (x, y)$ verify $0 \leq x < 1$, there exists K_A sufficiently large such that for all $k \geq K_A$, $||S_{k+1} - S_k||_\infty \leq \epsilon$. Indeed, let X denote the maximum value of x for all cycles and Y the maximum $|y|$. Then, for $k \geq \frac{\log(Y/\epsilon)}{\log(1/X)}$, $||S_{k+1} - S_k||_\infty \leq \epsilon$ for all (x, y). This leads us to consider an approximate version of the generic single-source shortest distance algorithm in non-acyclic cases, where the equality test is replaced by an ϵ-equality: $u =_\epsilon v$ if $||u - v||_\infty \leq \epsilon$. The following gives the pseudocode of the modified algorithm.

```
1  for i ← 1 to |Q|
2      do  d[i] ← r[i] ← 0̄
3  d[s] ← r[s] ← 1̄
4  S ← {s}
5  while  S ≠ ∅
6          do  q ← head(S)
7              DEQUEUE(S)
8              r′ ← r[q]
9              r[q] ← 0̄
10             for each e ∈ E[q]
11             do  if d[n[e]] ≠ε d[n[e]] ⊕ (r′ ⊗ w[e])
12                     then  d[n[e]] ← d[n[e]] ⊕ (r′ ⊗ w[e])
13                           r[n[e]] ← r[n[e]] ⊕ (r′ ⊗ w[e])
14                           if n[e] ∉ S
15                             then  ENQUEUE(S, n[e])
```

$d[q]$ denotes the tentative shortest distance from the source s to q. $r[q]$ keeps track of the sum of the weights added to $d[q]$ since the last queue extraction of q. The attribute r is needed for the shortest-distance algorithm to work in non-idempotent cases. The algorithm uses a queue S to store the set of states to consider for the relaxation steps of lines 11-15 [Mohri, 2002b]. Any queue discipline, e.g., FIFO, shortest-first, topological (in the acyclic case), can be used. The test of line 11 is based on an ϵ-equality.

Different queue disciplines yield different running times for our algorithm. The choice of the best queue discipline to use can be based on the structure of the two automata, which can be exploited to obtain a more efficient algorithm to compute the relative entropy. More specifically, let Q, E denote (respectively) the set of states and edges in the intersection automata. Further, let $N(q)$ denote the number of times a state q is inserted in the queue. Then, using the Fibonacci heap with a shortest first queue discipline (as in Dijkstra's algorithm), the complexity of the algorithm is given by:

$$O(|Q| + |E| \max_{q \in Q} N(q) + \log |Q| \sum_{q \in Q} N(q)). \tag{22}$$

If the underlying automata are acyclic, then using the queue discipline corresponding to the topological order yields the best time complexity, and the problem can be solved in linear time: $O(|Q| + |E|)$.

Using a breadth-first queue discipline (as in the Bellman-Ford shortest distance algorithm), updates to the shortest distance estimates in iteration k can be formulated as $D^k = MD^{k-1}$, where M is the *matrix associated to the automaton*, that is the matrix representing the weighted graph defined by the automaton. Note that the matrix multiplication here is over the $\oplus$ and $\otimes$ operations of the semiring, so that $D^k[i] = \oplus_{j=1}^{|Q|} M[i,j] \otimes D^{k-1}[j]$.

We now analyze the convergence rate of the approximate algorithm with the breadth-first queue discipline. Let us focus only on the first component of the distance pair. Let M_1 be the matrix obtained by taking the first part of each element of M. Assume that the matrix M is a stochastic matrix.

By the Perron-Frobenius theorem, we know that the largest eigenvalue is 1 and has a multiplicity of 1. Furthermore, all other eigenvalues λ are such that $|\lambda| < 1$. Using the Jordan canonical form of M, it is not hard to show that the matrix multiplication operation converges in $O(|\lambda_2|^k)$, where λ_2 is the second largest eigenvalue of M (see [Golub and Loan, 1996] for a similar analysis). Thus, the updates in the kth iteration are proportional to λ_2^k, hence, $k = \frac{\log(1/\epsilon)}{\log(1/|\lambda_2|)}$. Plugging in this expression for $N(q)$, the overall complexity of the approximate algorithm is:

$$O\left(|Q| + (|E| + |Q|)\frac{\log(1/\epsilon)}{\log(1/|\lambda_2|)}\right). \tag{23}$$

For ϵ exponentially smaller than $|\lambda_2|$ ($\epsilon = |\lambda_2|^d$), the cost in complexity is only linear: $O(|Q| + d(|E| + |Q|))$.

It is possible to use different queue disciplines in different parts of the graph and improve the running time of the algorithm. For example, for a large graph with several strongly connected components, one can use a topological order on the component graph, with shortest-first queue discipline in each strongly connected component [Mohri, 2002b]. If there are k strongly connected components, with the ith component having n_i vertices, then the running time is given by $O(|Q| + |E| \max_{q \in Q} N(q) + \log |\max_i n_i| \sum_{q \in Q} N(q))$. If the largest component has $O(n/k)$ vertices, then this improves the general complexity by an additive factor of $\sum_{q \in Q} N(q) \log k$. Our experience with such computations for very large graphs of several million states shows that the generic topological order with the shortest-first queue discipline within each strongly connected component often leads to the most efficient results in practice.

4.4. *Comparison with Previous Work*

In [Carrasco, 1997], the author describes a *procedure* for an approximate computation of the relative entropy of two deterministic stochastic automata. The procedure is based on an iterative method (which can be viewed as approximating the inverse of a matrix) for computing, for a stochastic automaton A, the probability of each state q, that is the sum of the weights of all paths going through q. The convergence is claimed but not proved and no bound is indicated on the maximum number of iterations.

The author reports no complexity result for the procedure described, which makes it difficult to compare with our algorithm. Our most favorable estimate of its complexity is $\Omega(|A|^2|B|^2(T + |\Sigma|))$, where T denotes the maximum number of iterations executed. This is because the procedure requires using a matrix of size $|A|^2|B|^2$. The complexity of the procedure also depends on the size of the alphabet, which, in some applications such as natural language processing applications, may be very large. Furthermore, the lower bound space complexity of this procedure is $\Omega(|A|^2|B|^2)$. This makes it unsuitable for computing the relative entropy of large weighted automata. Note that the experiments reported by the author were carried out with very small grammars of about 30 rules. Nevertheless, the procedure bears some resemblance with our approximate algorithm. It can be viewed as an alphabet-dependent non-sparse implementation of that algorithm for the particular case of a FIFO queue discipline.

4.5. *Experiments*

We implemented both the generic Floyd-Warshall algorithm and the approximate algorithm for the computation of the relative entropy of unambiguous probabilistic automata.

To avoid the numerical instability issues related to the multiplications of probabilities, we used instead negative log probabilities. This corresponds to taking the image of the entropy semiring by the semiring morphism $\log \times I$ where I is the identity over the second element of the weights.

To evaluate the efficiency of our approximate algorithm for computing the relative entropy we created two n-gram statistical models trained on a large corpus – one a bigram model ($n = 2$) and one a trigram model ($n = 3$). The minimal deterministic weighted automaton representing the bigram model had about 200,000 transitions, that of the trigram model about 400,000 transitions. It took about 3s on a single 2GHz Intel processor with 128MB of RAM to compute the relative entropy of these large weighted automata using a FIFO queue discipline. With a shortest-first queue discipline, the time was reduced to 2s.

5. Relative Entropy of Arbitrary Probabilistic Automata

This section proves a hardness result suggesting that the problem of computing the relative entropy of arbitrary probabilistic automata is intractable.

5.1. *Hardness Result*

We describe a reduction of the problem of determining whether the language accepted by an automaton is Σ^* to the that of determining whether the relative entropy of two probabilistic automata is infinite.

Automaton A_0. We first describe an automaton A_0 that is used in our reduction. Fix a real number $\alpha > 0$ such that $\alpha|\Sigma| < 1$ and let A_0 be the one-state weighted automaton representing the weighted regular expression $(1-\alpha)(\sum_{x\in\Sigma} \alpha x)^*$ shown in Figure 3 for $\Sigma = \{a, b\}$. By definition, A_0 accepts all strings $x \in \Sigma^*$ and for all $x \in \Sigma^*$, $[\![A]\!](x) = \alpha^{|x|}(1 - |\Sigma|\alpha)$. By construction, A_0 is stochastic and thus probabilistic. Here is also a direct verification:

$$\sum_{x\in\Sigma^*} [\![A_0]\!](x) = \sum_{n=0}^{\infty} \sum_{|x|=n} \alpha^n(1 - |\Sigma|\alpha) = \sum_{n=0}^{\infty} |\Sigma|^n \alpha^n(1 - |\Sigma|\alpha) \tag{24}$$

$$= (1 - |\Sigma|\alpha)\frac{1}{1 - |\Sigma|\alpha} = 1. \tag{25}$$

The following theorem shows that the problem of determining the relative entropy of two arbitrary probabilistic automata is at least as hard as determining if a finite automaton accepts Σ^*.

Theorem 9. *Let A be an arbitrary probabilistic automaton, then $D(A_0\|A) < \infty$ iff A accepts Σ^*.*

Proof. Assume that $[\![A]\!](x) = 0$ for some $x \in \Sigma^*$. Then, since $[\![A_0]\!](x) > 0$, $[\![A_0]\!](x) \log \frac{[\![A_0]\!](x)}{[\![A]\!](x)}$ is infinite and $D(A_0\|A) = \infty$.

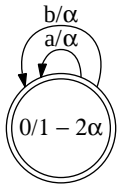

Fig. 3. The automaton A_0 that accepts all strings, $\{a, b\}^*$, and assigns a weight of $\alpha^n(1-\alpha)$ to any string of length n. $\alpha > 0$ is a constant such that $2\alpha < 1$.

Assume now that A accepts Σ^*, thus $[\![A]\!](x) \neq 0$ for all $x \in \Sigma^*$. Without loss of generality, we can assume A to be trim. Let E denote the set of transitions of A and let δ denote the minimum weight of a transition: $\delta = \min_{e \in E} w[e]$. By assumption, $\delta > 0$ since the automaton A is trim and probabilistic. For $x \in \Sigma^*, |x| = n$, $[\![A]\!](x) \geq \delta^n$. Thus

$$\forall x \in \Sigma^*,\ \frac{[\![A_0]\!](x)}{[\![A]\!](x)} = \frac{\alpha^n(1-|\Sigma|\alpha)}{[\![A]\!](x)} \leq (1-|\Sigma|\alpha)\left(\frac{\alpha}{\delta}\right)^n. \tag{26}$$

It follows that:

$$\forall x \in \Sigma^*,\ [\![A_0]\!](x) \log \frac{[\![A_0]\!](x)}{[\![A]\!](x)} \leq \alpha^n(1-|\Sigma|\alpha)\left(n\log(\alpha/\delta) + \log(1-|\Sigma|\alpha)\right). \tag{27}$$

For any positive integer N, summing over all strings x of length at most N, in the order of increasing $|x|$ yields:

$$\begin{aligned}\sum_{|x|\leq N} [\![A_0]\!](x) \log \frac{[\![A_0]\!](x)}{[\![A]\!](x)} &= \sum_{n=0}^{N} \sum_{x:|x|=n} [\![A_0]\!](x) \log \frac{[\![A_0]\!](x)}{[\![A]\!](x)} \\ &\leq \sum_{n=0}^{N} |\Sigma|^n \alpha^n (1-|\Sigma|\alpha)\left(n\log(\alpha/\delta) + \log(1-|\Sigma|\alpha)\right).\end{aligned} \tag{28}$$

Since $\alpha|\Sigma| < 1$ the two series in this summation, $\sum_n n\beta^n$ and $\sum_n \beta^n$ with $\beta = |\Sigma|\alpha < 1$, converge. It is straightforward to verify that for $0 \leq \beta < 1$, $\sum_{n=0}^{\infty} n\beta^n = \frac{\beta}{(1-\beta)^2}$. Using this identity, we obtain the following bound on $D(A_0\|A)$:

$$D(A_0\|A) \leq (1-|\Sigma|\alpha)\left(\frac{|\Sigma|\alpha\log(\alpha/\delta)}{(1-|\Sigma|\alpha)^2} + \frac{\log(1-|\Sigma|\alpha)}{1-|\Sigma|\alpha}\right). \tag{29}$$

Thus $D(A_0\|A) < \infty$. □

Theorem 10. *The problem of computing the relative entropy of two arbitrary probabilistic automata is PSPACE-complete.*

Proof. The universality problem, i.e., the problem of deciding if a trim finite automaton A accepts Σ^*, is PSPACE-complete [Stockmeyer and Meyer, 1973, Garey

and Johnson, 1979]. The transitions of any trim finite automaton A can be augmented with positive weights so that it becomes a probabilistic automaton. This can be done by weighting each outgoing transition of state q, or final weight if q is final, by $1/n_q$ where n_q is the out-degree of q, plus one if q is final. The encoding of $1/n_q$ takes $O(\log_2 n_q)$ space, thus the space and time complexity of this construction is polynomial in the size of A. By Theorem 9, it can be decided if a probabilistic automaton A accepts all strings by computing the relative entropy $D(A_0\|A)$ and testing its finiteness. Thus, the computation of the relative entropy can determine if a trim finite automaton A accepts Σ^*. □

5.2. *Remarks*

Theorem 10 suggests that the general problem of computing the relative entropy of arbitrary probabilistic automata is intractable. However, one may resort to various approximations of practical importance. An example is an approximation based on the use of the log-sum inequality by [Singer and Warmuth, 1997] in the context of machine learning. We have initiated a specific study of such approximations, in particular by examining the quality of an approximation when using the algorithms we presented for the unambiguous case.

Note that the general problem of determining if a weighted automaton over the $(\mathbb{R}, +, \cdot, 0, 1)$ semiring accepts the full free monoid Σ^* is undecidable [Berstel and Reutenauer, 1988]. Here, we are considering the same decidability question but only for probabilistic automata, which form a restricted class of all weighted automata over the $(\mathbb{R}, +, \cdot, 0, 1)$ semiring. However, we conjecture that the problem is in fact undecidable even in this case.

6. Relative Entropy as a Kernel

This section examines the use of the relative entropy, or its symmetrized version, in machine learning algorithms. The results hold in general and are not limited to the particular case of probabilistic automata.

In machine learning, functions $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are called *kernels*. A kernel is said to be *positive definite symmetric* (PDS for short) if it is *symmetric*, $K(x, y) = K(y, x)$ for all $x, y \in \mathcal{X}$, and if for any subset $\{x_1, \ldots, x_m\} \subseteq \mathcal{X}$, the eigenvalues of the matrix $[K(x_i, x_j)]_{1 \leq i,j \leq m}$ are non-negative. PDS kernels play an important role in machine learning since they can be combined with discriminant algorithms such as support vector machines (SVMs) to create powerful predictors [Schölkopf and Smola, 2002], the PDS condition ensuring the convergence of training.

In some cases, a symmetric kernel K is not positive definite but $\exp(-\lambda K)$ is PDS for any $\lambda > 0$. K is then said to be *negative definite symmetric* (NDS). Such kernels are also important since they can be used to defined PDS kernels as in the case of Gaussian kernels.

We will show however that the symmetrized relative entropy is neither PDS nor NDS, contrarily to what is stated in a number of machine learning papers, which

limits its use and application in kernel methods.

The *symmetrized relative entropy* of two distributions p and q is given by:

$$D_{sym}(p\|q) = \frac{D(p\|q) + D(q\|p)}{2} = \sum_{x\in\mathcal{X}} [p(x) - q(x)] \log \frac{p(x)}{q(x)}. \tag{30}$$

Theorem 11. *The symmetrized relative entropy is not a PDS kernel.*

Proof. Let $\{q_1, q_2, \ldots, q_m\}$ be a set of probability distributions over $\mathcal{X}$. Consider the Gram matrix $\mathbf{K} \in \mathbb{R}^{m\times m}$ defined by $\mathbf{K}_{i,j} = D_{sym}(q_i\|q_j)$. By definition of D_{sym}, $D_{sym}(q_i\|q_i) = 0$ for all $i \in [1, m]$, thus $\mathbf{tr}(\mathbf{K}) = 0$. When $\mathbf{K} \neq 0$, this implies that $\mathbf{K}$ admits at least one negative eigenvalue. □

To show that the symmetrized relative entropy is not an NDS kernel, we use the following theorem of Schoenberg [1938].

Theorem 12 ([Schoenberg, 1938, Berg et al., 1984]) *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be an NDS kernel such that for $x, y \in \mathcal{X}$, $K(x, y) = 0$ iff $x = y$. Then, there exist a Hilbert space H and a mapping $\Phi : \mathcal{X} \to H$ such that*

$$\forall x, y \in \mathcal{X},\ K(x, y) = ||\Phi(x) - \Phi(y)||^2. \tag{31}$$

Thus, under the hypothesis of the theorem, $\sqrt{K}$ defines a metric.

Theorem 13. *The symmetrized relative entropy is not an NDS kernel.*

Proof. Note that for any two distributions p and q, $D_{sym}(p\|q) = 0$ iff $D(p\|q) = D(q\|p) = 0$ that is iff $p = q$. Thus, by Theorem 12, if D_{sym} is an NDS kernel, $\sqrt{D_{sym}}$ defines a metric. We prove that $\sqrt{D_{sym}}$ does not obey the triangle inequality, which will show that D_{sym} is not NDS.

For the sake of simplicity, the proof is given in the case of a universe of events limited to two elements: $\mathcal{X} = \{x_1, x_2\}$. Let $\epsilon > 0$ and let q_1, q_2, q_3 be the three distributions over $\mathcal{X}$ defined by:

$$\forall i \in [1, 3],\ q_i(x_1) = 1 - i\epsilon \text{ and } q_i(x_2) = i\epsilon. \tag{32}$$

By definition of the symmetrized relative entropy,

$$D_{sym}(q_1\|q_2) = \epsilon \log \frac{1-\epsilon}{1-2\epsilon} - \epsilon \log \frac{\epsilon}{2\epsilon} = \epsilon \log \frac{2(1-\epsilon)}{1-2\epsilon}. \tag{33}$$

Similarly, $D_{sym}(q_2\|q_3) = \epsilon \log \frac{3(1-2\epsilon)}{2(1-3\epsilon)}$ and $D_{sym}(q_1\|q_3) = 2\epsilon \log \frac{3(1-2\epsilon)}{1-3\epsilon}$. Note that:

$$\begin{aligned} D_{sym}(q_1\|q_3) &= 2\epsilon \log \tfrac{3(1-2\epsilon)}{1-3\epsilon} = 2\big(\epsilon \log \tfrac{2(1-\epsilon)}{1-2\epsilon} + \epsilon \log \tfrac{3(1-2\epsilon)}{2(1-3\epsilon)}\big) \\ &= 2\big(D_{sym}(q_1\|q_2) + D_{sym}(q_2\|q_3)\big). \end{aligned} \tag{34}$$

Since $\sqrt{\cdot}$ is strictly concave,

$$\begin{aligned} \sqrt{D_{sym}(q_1\|q_3)} &= 2\sqrt{\tfrac{D_{sym}(q_1\|q_2)}{2} + \tfrac{D_{sym}(q_2\|q_3)}{2}} \\ &> \sqrt{D_{sym}(q_1\|q_2)} + \sqrt{D_{sym}(q_2\|q_3)}. \end{aligned} \tag{35}$$

This shows that $\sqrt{D_{sym}}$ does not obey the triangle inequality. □

7. Computation of the Norm of a Probabilistic Automaton

In Section 4, we gave a general algorithm for computing the relative entropy of two unambiguous probabilistic automata by relating this problem to a shortest-distance problem over the appropriate semiring. A special case of that algorithm can be used to compute the entropy of a single unambiguous probabilistic automaton. One may ask if such results could be generalized to the computation of other similar quantities that we will refer to as the *norm of an unambiguous probabilistic automaton*. This section shows how they can be generalized indeed by considering an arbitrary monoid morphism.

7.1. *Computation of the Norm of an Unambiguous Probabilistic Automaton*

Let $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ be a closed semiring, or an ϵ-k-closed semiring for an automaton A. Let $\Phi : (\mathbb{R}_+, \cdot, 1) \to (\mathbb{K}, \otimes, \overline{1})$ be a monoid morphism. We will say that Φ *preserves closedness*, if for all x, $0 \leq x < 1, \bigoplus_{n=0}^{\infty} \Phi(x^n)$ is well-defined and in $\mathbb{K}$. For a such a morphism, we can define the Φ-*norm of a probabilistic automaton* as:

$$\|A\|_\Phi = \bigoplus_{x \in \Sigma^*} \Phi([\![A]\!](x)). \tag{36}$$

Theorem 14. *Let $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ be a closed or ϵ-k-closed semiring and let $\Phi : (\mathbb{R}_+, \cdot, 1) \to (\mathbb{K}, \otimes, \overline{1})$ be a monoid morphism preserving closedness. Then, for any unambiguous probabilistic automaton A, $\|A\|_\Phi$ can be computed exactly in time $O(|A|^3)$.*

Proof. The automaton $\Phi(A)$ derived from A by replacing each weight x by $\Phi(x)$ is a weighted automaton over the semiring $\mathbb{K}$. Since A is unambiguous, at most one successful path in A, $\pi = e_1 \cdots e_k$, is labeled with any string $x \in \Sigma^*$. Since Φ is a monoid morphism, $\Phi([\![A]\!](x)) = \bigotimes_{j=1}^{k} \Phi(w[e_j])$, that is the weight of the path labeled with x in $\Phi(A)$. This shows that $\|A\|_\Phi = s(A)$ and proves the theorem. □

Theorem 14 provides an algorithm for computing the Φ-norm of unambiguous probabilistic automata for arbitrary monoid morphisms preserving closedness. We will briefly illustrate two applications of the theorem.

(a) Entropy of a Probabilistic Automaton.

Let $(\mathbb{K}, \oplus, \otimes, (0,0), (1,0))$ be the entropy semiring. It is not hard to see that function $\Phi : (\mathbb{R}_+, +, \cdot, 0, 1) \to (\mathbb{K}, \oplus, \otimes, (0,0), (1,0))$ defined by: $\forall x \in \mathbb{R}_+, \Phi(x) = (x, -x \log x)$, is a monoid morphism preserving closedness. Thus, the norm-Φ of an unambiguous probabilistic automaton can be computed efficiently using a single-source shortest-distance algorithm. Its second component is exactly the entropy of A, thus this provides an

efficient and simple algorithm for computing the entropy of A.

(b) Norm L_α of a Probabilistic Automaton, $\alpha \in \mathbb{R}_+$.

The function $\Phi : (\mathbb{R}_+, +, \cdot, 0, 1) \to (\mathbb{R}_+, +, \cdot, 0, 1)$ defined by $\Phi(x) = x^\alpha$ is clearly a monoid morphism. Since for $0 \leq x < 1$, $0 \leq x^\alpha < 1$, it also preserves closedness. Thus, the L_α-norm of an unambiguous probabilistic automaton A can be computed efficiently using a shortest-distance algorithm. In particular, the Bhattacharya norm, i.e., $L_{\frac{1}{2}}$-norm, of A can be computed efficiently.

7.2. *Computation of the Norm of Arbitrary Automata*

In general, a probabilistic automaton may not be unambiguous. But, the L_p-norm can still be computed in polynomial time for any integer $p \geq 1$.

Theorem 15. *The L_p-norm of a probabilistic automaton A can be computed exactly in time $O(|A|^{3p})$ time and $\Theta(|A|^{2p})$ space.*

Proof. Let $A^{(p)}$ denote the automaton obtained by intersecting A with itself $p-1$ times. Then, by definition of intersection, $(s[A^{(p)}])^{1/p}$ represents the L_p-norm of A. The cost of intersection to create $A^{(p)}$ is in $O(|A|^p)$. □

Note that the problem of computing the L_∞ norm of a probabilistic is NP-hard [Rune B. Lyngsø and Christian N. S. Pederson, 2002].

7.3. *Approximate Computation*

Here we consider the specific case of the computation of the L_p-norm of a probabilistic automaton. Our results can be generalized to cover more general cases, in particular in the case of unambiguous automata.

Since for any $\epsilon > 0$, a probabilistic automaton is ϵ-k-closed for the probability semiring, instead of the (generalized) Floyd-Warshall algorithm, we can use a single-source shortest-distance algorithm to compute $s[A]$ as already described in Section 4.3. This algorithm works with any queue discipline and its space complexity is linear which is significantly more efficient than the Floyd-Warshall algorithm. The complexity results and analyses detailed in Section 4.3 apply identically here.

8. Conclusion

We presented an exhaustive study of the problem of computing the relative entropy of probabilistic automata.

Our results demonstrate the benefit of semiring theory for the formulation of the problem which becomes as a single-source shortest-distance one. This results in the

definition of simple but efficient algorithms, both exact and approximate, for the computation of the relative entropy of two unambiguous probabilistic automata or the entropy of a single unambiguous probabilistic automaton. As shown by our experimental results, these algorithms scale to large probabilistic automata of several hundred thousand transitions.

Our algorithms can be adapted straightforwardly to compute the so-called unnormalized relative entropy of two unambiguous weighted automata over $(\mathbb{R}_+, +, \cdot, 0, 1)$, defined by:

$$D(A\|B) = \sum_x [\![A]\!](x) \log \frac{[\![A]\!](x)}{[\![B]\!](x)} - [\![A]\!](x) + [\![B]\!](x) \tag{37}$$

simply by replacing Φ_1 and Φ_2 by Φ'_1 and Φ'_2, where $\Phi'_1(A)$ ($\Phi'_2(A)$) is the weighted automaton over the entropy semiring derived from A by replacing each weight w with the pair $(w, 1)$ (resp (w, w)). The entropy semiring can also be used to give a conceptually simple formulation of the computation of the relative entropy of tree automata and to derive similar computation algorithms.

We proved that the computation of the relative entropy of arbitrary probabilistic automata is PSPACE-complete and thus likely to be intractable. This suggests examining approximate computations of the relative entropy. We have already initiated the study of a natural approximate computation of the relative entropy that extends the results presented in this paper.

Acknowledgments

The work of Mehryar Mohri and Ashish Rastogi was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR). This project was also sponsored in part by the Department of the Army Award Number W23RYX-3275-N605. The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702-5014 is the awarding and administering acquisition office. The content of this material does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

References

Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag: Berlin-New York, 1984.

Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*. Springer-Verlag: Berlin-New York, 1988.

Stephen Bloom and Zoltan Ésik. *Iteration Theories*. Springer-Verlag, Berlin, 1991.

Rafael C. Carrasco. Accurate computation of the relative entropy between stochastic regular grammars. *Informatique Théorique et Applications (ITA)*, 31(5):437–444, 1997.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press: Cambridge, MA, 1992.

Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. On the Computation of Some Standard Distances between Probabilistic Automata. In *Proceedings of the 11th International Conference on Implementation and Application of Automata (CIAA 2006)*, volume 4094 of *Lecture Notes in Computer Science*, pages 137–149, Taipei, Taiwan, August 2006. Springer-Verlag, Heidelberg, Germany.

Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. L_p Distance and Equivalence of Probabilistic Automata. *International Journal of Foundations of Computer Science*, to appear, 2007.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

Imre Csiszar and Janos Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Akademiai Kiado, 1997.

Karel Culik II and Jarkko Kari. Digital Images and Formal Languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 599–616. Springer, 1997.

R. Durbin, S.R. Eddy, A. Krogh, and G.J. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge UK, 1998.

Samuel Eilenberg. *Automata, Languages and Machines*, volume A–B. Academic Press, 1974–1976.

Jason Eisner. Expectation Semirings: Flexible EM for Finite-State Transducers. In *Proceedings of the ESSLLI Workshop on Finite-State Methods in NLP*, 2001.

Michael R. Garey and David S. Johnson. *Computers and Intractability*. Freeman and Company, New York, 1979.

G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.

Anders Krogh, Michael Brown, I. Saira Mian, Kimmen Sjolander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–1531, 1994.

Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*. Number 5 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, Germany, 1986.

Daniel J. Lehmann. Algebraic Structures for Transitive Closures. *Theoretical Computer Science*, 4:59–76, 1977.

Michael Mandel, Graham Poliner, and Dan Ellis. Support vector machine active learning for music retrieval. *Multimedia Systems*, 12(1):3–13, 2006.

Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2), 1997.

Mehryar Mohri. Generic Epsilon-Removal and Input Epsilon-Normalization Algorithms for Weighted Transducers. *International Journal of Foundations of Computer Science*, 13(1):129–143, 2002a.

Mehryar Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002b.

Mehryar Mohri. General Algebraic Frameworks and Algorithms for Shortest-Distance Problems. Technical Memorandum 981210-10TM, AT&T Labs - Research, 62 pages, 1998.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted Automata in Text and Speech Processing. In *Proceedings of the 12th biennial European Conference on Artificial Intelligence (ECAI-96), Workshop on Extended finite state models of language, Budapest, Hungary*. John Wiley and Sons, Chichester, 1996.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted Finite-State Transducers in Speech Recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

Azaria Paz. *Introduction to probabilistic automata*. Academic Press, New York, 1971.

Michael O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

Rune B. Lyngsø and Christian N. S. Pederson. The Consensus String Problem and the Complexity of Comparing Hidden Markov Models. *Journal of Computer and System Sciences*, 65(3):545–569, 2002.

Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag, 1978.

I. J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, november 1938.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels*. MIT Press: Cambridge, MA, 2002.

Yoram Singer and Manfred K. Warmuth. Training Algorithms for Hidden Markov Models using Entropy Based Distance Functions. In *Advances in Neural Information Processing Systems*, volume 9, page 641. The MIT Press, 1997.

L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proceedings of the* 5^{th} *Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, New York, 1-9., 1973.

Flemming Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Trans. Inform. Theory*, 46:1602–1609, 2000.