Efficient Computation of the Relative Entropy of Probabilistic Automata

Corinna Cortes[1], Mehryar Mohri[2,1] *, Ashish Rastogi[2], and Michael D. Riley[1]

[1] Google Research, New York, NY, USA.
[2] Courant Institute of Mathematical Sciences,
New York University,
New York, NY, USA.

Abstract. The problem of the efficient computation of the relative entropy of two distributions represented by deterministic weighted automata arises in several machine learning problems. We show that this problem can be naturally formulated as a *shortest-distance problem* over an intersection automaton defined on an appropriate semiring. We describe simple and efficient novel algorithms for its computation and report the results of experiments demonstrating the practicality of our algorithms for very large weighted automata. Our algorithms apply to *unambiguous* weighted automata, a class of weighted automata that strictly includes *deterministic* weighted automata. These are also the first algorithms extending the computation of entropy or of relative entropy beyond the class of deterministic weighted automata.

1 Introduction

The relative entropy, or Kullback-Leibler divergence, is used in a variety of contexts as a measure of the discrepancy of two distributions p and q [5]. It is an asymmetric difference that, from the point of view of coding theory, measures the number of additional bits needed to encode p, when using an optimal code for q in place of an optimal code for p.

The problem of the efficient computation of the relative entropy of two distributions represented by weighted automata arises in several machine learning problems. Weighted automata are used extensively in text and speech processing to model different aspects of language such as morphology, phonology, or syntax [12]. The output of a large-vocabulary speech recognition system or that of a complex information extraction system is typically represented as a weighted automaton compactly representing a large set of alternative sequences [17]. Weighted automata are also used in other applications such as image processing [6].

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 relative entropy of the automaton inferred and that of the target automaton. Similarly, in some grammar inference applications, the convergence of an iterative algorithm relies on the magnitude of the relative entropy of two consecutive weighted automata. The relative entropy is also often used for clustering large sets of automata, such as those output by a speech recognition or information extraction system.

* This work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR).

This motivates the design of efficient algorithms for the computation of the relative entropy of two weighted automata. One approximate solution 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 a detailed analysis of the problem of the computation of the relative entropy of weighted automata in the case where they are *deterministic* or, more generally, *unambiguous*, i.e., no two successful paths are labeled with the same string. We show that the problem can be formulated naturally as a *single-source shortest-distance problem* over an intersection automaton defined on an appropriate semiring that we will refer to as the *entropy semiring*. We describe simple and efficient algorithms for the computation of relative entropy and report the results of experiments demonstrating the practicality of our algorithms for very large weighted automata.

A procedure for the approximate computation of the relative entropy was given by [3]. 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 [3] [eq. 15, 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 [3] and have a better time and space complexity.

The paper is organized as follows. Section 2 introduces the preliminary semiring and automata definitions used in the remaining of the paper. Section 3 introduces the entropy semiring and formulates the computation of the relative entropy in terms of shortest-distances over that semiring. Section 4 describes both an exact and a fast approximate algorithm for the computation of the relative entropy. Section 5 briefly reports the results of experiments demonstrating the practicality of our algorithms for very large weighted automata.

2 Preliminaries

Weighted automata are automata in which each transition carries some weight in addition to the usual alphabet symbol [7, 18, 1]. For various operations to be well-defined, the weight set must have the algebraic structure of a semiring [10]. 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* $\mathbb{K}$*:* $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \quad$ *and* $\quad c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$*, and* $\overline{0}$ *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 [14]. Furthermore, it is assumed that the automata are *trim*, i.e. all states in the automata are both accessible and co-accessible.

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 $p[\pi] = p[e_1]$. 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:

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

Our algorithms for the computation of the entropy of a weighted automata or the computation of the relative entropy of two automata apply to *unambiguous weighted automata*. A weighted automaton is said to be *unambiguous* if for any $x \in \Sigma^*$ it admits at most one accepting path labeled with x. Thus, the class of unambiguous weighted automata includes *deterministic* 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.

Fig. 1 (a) shows an unambiguous weighted automaton that does not admit an equivalent deterministic weighted automaton (the proof will be included in a future journal version). Previous work on the computation of the relative entropy [3] was limited to deterministic finite automata. We present the first algorithms for the computation of the relative entropy of unambiguous weighted automata.

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 $P(p,q) = \emptyset$. Let A be a weighted automaton defined over the probability semiring $(\mathbb{R}_+, +, \times, 0, 1)$. We will say that A is *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 $\mathbb{K}$ and $\sum_{x \in \Sigma^*} [\![A]\!](x) = 1$. *Stochastic automata* are probabilistic automata such that at each state the weights of the outgoing transitions and the final weight sum to one.

Let A_1 and A_2 be two weighted automata 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 presence of ϵ-transitions [16]. 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 Formulation of the Problem

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 Relative entropy

The entropy $H(p)$ of a probability mass function p defined over a discrete set $\mathcal{X}$ is defined as [5]:

$$H(p) = -\sum_{x \in \mathcal{X}} p(x) \log p(x), \tag{3}$$

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

$$D(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathrm{E}_p[\log \frac{p(X)}{q(X)}], \tag{4}$$

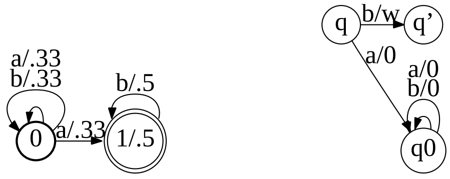
where we use the standard conventions: $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$. It is easy to show using Jensen's inequality and the concavity of the log function that the relative entropy is a non-negative number and that $D(p\|q) = 0$ if and only if $p = q$. Note that $D(p\|q)$ is not a metric because it is not symmetric and does not satisfy the triangle inequality.

These definitions can be naturally extended to probabilistic automata which define distributions over sets of strings. The relative entropy of A and B can be written as the sum of two terms:[3]

$$D(A\|B) = \sum_x [\![A]\!](x) \log [\![A]\!](x) - \sum_x [\![A]\!](x) \log [\![B]\!](x). \tag{5}$$

The next section introduces a semiring, the *entropy semiring*, showing that each term can be viewed as a single-source shortest-distance for an automaton defined over that semiring.

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



(a) (b)

Fig. 1. (a) 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 $(a^*b^*)^*ab^*$. (b) Illustration of the completion operation.

3.2 Entropy semiring

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 :

$$(x_1, y_1) \oplus (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \tag{6}$$

$$(x_1, y_1) \otimes (x_2, y_2) = (x_1 x_2, x_1 y_2 + x_2 y_1) \tag{7}$$

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

Proof. The proof is rather straightforward and will be included in the journal version. □

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 contexts and can be defined in terms of an S-module [2, 8].

3.3 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 1 (b)). 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 1. *The relative entropy of A and B satisfies the following identity in the entropy semiring:*

$$(0, D(A\|B)) = s[\Phi_1(A) \cap \Phi_2(\log A)] - s[\Phi_1(A) \cap \Phi_2(\log B)]. \tag{8}$$

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

$$([\![A]\!](x), 0) \otimes (1, \log[\![A]\!](x)) = ([\![A]\!](x), [\![A]\!](x) \log[\![A]\!](x)). \tag{9}$$

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

$$\begin{aligned} s[\Phi_1(A) \cap \Phi_2(\log A)] &= \bigoplus_x ([\![A]\!](x), [\![A]\!](x) \log[\![A]\!](x)) & (10)\\ &= (\sum_x [\![A]\!](x), \sum_x [\![A]\!](x) \log[\![A]\!](x)) & (11)\\ &= (1, \sum_x [\![A]\!](x) \log[\![A]\!](x)). & (12) \end{aligned}$$

Similarly, we can show that

$$s[\Phi_1(A) \cap \Phi_2(\log B)] = (1, \sum_x [\![A]\!](x) \log[\![B]\!](x)). \tag{13}$$

The statement of the proposition follows directly from the identities 12 and 13 and Equation 5. □

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.

4 Algorithms

This section describes two algorithms for computing a single-source shortest distance over the entropy semiring, an exact algorithm, and a more efficient and more practical approximate algorithm.

4.1 Exact solution

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 [13, 15]. 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 [11] that is more general than the classical one used by Cormen *et al.* [4] 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 [13, 15]. More generally, given a graph or automaton A, we introduce the following definition.

Definition 3. *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 2. *Let A be a weighted automaton over the entropy semiring such that for any cycle weight $w = (x, y)$, $0 \leq x < 1$. Then, the entropy semiring is closed for A.*

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

$$R_k = \overbrace{(x, y) \otimes \ldots \otimes (x, y)}^{k \text{ times}}. \tag{14}$$

with $R_0 = (1, 0)$. We can show by induction that $R_k = (x^k, kyx^{k-1})$. The base case is readily established for $k = 0$. Assume that the hypothesis holds for all $i < k$. Then

$$\begin{aligned} R_k &= R_{k-1} \otimes (x, y) \\ &= (x^{k-1}, (k-1)yx^{k-2}) \otimes (x, y) \\ &= (x^k, kyx^{k-1}). \end{aligned} \tag{15}$$

For $N \geq 0$, define S_N by: $S_N = \bigoplus_{i=0}^{N} R_i$. It is easy to prove by induction as above that S_N verifies

$$S_N = \Big(\frac{1 - x^{N+1}}{1 - x}, y \cdot \big[\frac{1 - x^N}{(1 - x)^2} - \frac{Nx^N}{1 - x}\big]\Big). \tag{16}$$

Thus, for $0 \leq x < 1$, the closure of (x, y) is well-defined and in $\mathbb{K}$:[4]

$$(x, y)^* = \lim_{N \to \infty} S_N = (\frac{1}{1 - x}, \frac{y}{(1 - x)^2}). \tag{17}$$

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. □

[4] The right-hand side can 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 $(u, u \log u)$ (see Equation 9), where u is the weight of a 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 [13]; the following gives the corresponding pseudocode.

```
for $i \leftarrow 1$ to $|Q|$
    do for $j \leftarrow 1$ to $|Q|$
           do $d[i,j] \leftarrow \bigoplus_{e \in P(i,j)} w[e]$
for $k \leftarrow 1$ to $|Q|$
    do for $i \leftarrow 1$ to $|Q|$
           do for $j \leftarrow 1$ to $|Q|$
                  do $d[i,j] \leftarrow d[i,j] \oplus (d[i,k] \otimes d[k,k]^* \otimes d[k,j])$
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.

4.2 Approximate Solution

A generic single-source shortest-distance algorithm was presented for directed graphs defined over a *k-closed* semiring in [15]. 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 4. *Let $k \geq 0$ be an integer. A semiring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ is* k-closed *if:*

$$\forall a \in \mathbb{K}, \; \bigoplus_{n=0}^{k+1} a^n = \bigoplus_{n=0}^{k} a^n. \tag{18}$$

More generally, we will say that $\mathbb{K}$ is *k-closed for a graph G or automaton A*, if Equation 18 holds for all cycle weights $a \in \mathbb{K}$.

By definition, the entropy semiring is k-closed 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,

$$\forall k > 0, S_{k+1} - S_k = R_{k+1} = (x^{k+1}, (k+1)yx^k). \tag{19}$$

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 $i \leftarrow 1$ to $|Q|$
2      do $d[i] \leftarrow r[i] \leftarrow \overline{0}$
3  $d[s] \leftarrow r[s] \leftarrow \overline{1}$
4  $S \leftarrow \{s\}$
5  while $S \neq \emptyset$
6          do $q \leftarrow head(S)$
7              Dequeue($S$)
8              $r' \leftarrow r[q]$
9              $r[q] \leftarrow \overline{0}$
10             for each $e \in E[q]$
11             do if $d[n[e]] \neq_\epsilon d[n[e]] \oplus (r' \otimes w[e])$
12                   then $d[n[e]] \leftarrow d[n[e]] \oplus (r' \otimes w[e])$
13                        $r[n[e]] \leftarrow r[n[e]] \oplus (r' \otimes w[e])$
14                        if $n[e] \notin S$
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 [15]. 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:

$$O(|Q| + |E| \max_{q \in Q} N(q) + \log |Q| \sum_{q \in Q} N(q)). \tag{20}$$

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 [9] 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:

$$O(|Q| + (|E| + |Q|) \frac{\log(1/\epsilon)}{\log(1/|\lambda_2|)}). \tag{21}$$

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 [15]. 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-fist queue discipline within each strongly connected component often leads to the most efficient results in practice.

4.3 Comparison with previous work

In [3], 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.

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.

6 Conclusion

We described several algorithms for the computation of the relative entropy of two deterministic weighted automata or the entropy of a single deterministic weighted automaton by formulating the problem as a shortest-distance computation over the entropy semiring. We presented both an exact algorithm and an approximate algorithm that was shown to be very efficient even for very large automata of several hundred thousand transitions. The results demonstrate the benefit of a semiring-theory formulation of the problem. Our algorithms can be used similarly to compute the so-called unnormalized relative entropy of two weighted automata, which is defined by:

$$D(A\|B) = \sum_{x} [\![A]\!](x) \log \frac{[\![A]\!](x)}{[\![B]\!](x)} - [\![A]\!](x) + [\![B]\!](x) \tag{22}$$

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.

References

1. Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*. Springer-Verlag: Berlin-New York, 1988.
2. Stephen Bloom and Zoltan Ésik. *Iteration Theories*. Springer-Verlag, Berlin, 1991.
3. Rafael C. Carrasco. Accurate computation of the relative entropy between stochastic regular grammars. *Informatique Théorique et Applications (ITA)*, 31(5):437–444, 1997.
4. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press: Cambridge, MA, 1992.
5. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.
6. 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.
7. Samuel Eilenberg. *Automata, Languages and Machines*, volume A–B. Academic Press, 1974–1976.
8. Jason Eisner. Expectation Semirings: Flexible EM for Finite-State Transducers. In *Proceedings of the ESSLLI Workshop on Finite-State Methods in NLP*, 2001.
9. G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996.
10. Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*. Number 5 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, Germany, 1986.
11. Daniel J. Lehmann. Algebraic Structures for Transitive Closures. *Theoretical Computer Science*, 4:59–76, 1977.
12. Mehryar Mohri. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2), 1997.
13. Mehryar Mohri. General Algebraic Frameworks and Algorithms for Shortest-Distance Problems. Technical Memorandum 981210-10TM, AT&T Labs - Research, 62 pages, 1998.
14. Mehryar Mohri. Generic Epsilon-Removal and Input Epsilon-Normalization Algorithms for Weighted Transducers. *International Journal of Foundations of Computer Science*, 13(1):129–143, 2002.
15. Mehryar Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.
16. 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.
17. 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.
18. Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag, 1978.