Journal of Machine Learning Research 1 (2004) 1-50 Submitted 12/03; Published 01/04

Rational Kernels: Theory and Algorithms

Corinna Cortes CORINNA@GOOGLE.COM
Google Labs
1440 Broadway, New York, NY 10018

Patrick Haffner HAFFNER@RESEARCH.ATT.COM
AT&T Labs – Research
180 Park Avenue, Florham Park, NJ 07932

Mehryar Mohri MOHRI@RESEARCH.ATT.COM
AT&T Labs – Research
180 Park Avenue, Florham Park, NJ 07932

Editor: Kristin Bennett and Nicolò Cesa-Bianchi

Abstract

Many classification algorithms were originally designed for fixed-size vectors. Recent applications in text and speech processing and computational biology require however the analysis of variable-length sequences and more generally weighted automata. An approach widely used in statistical learning techniques such as Support Vector Machines (SVMs) is that of kernel methods, due to their computational efficiency in high-dimensional feature spaces. We introduce a general family of kernels based on weighted transducers or rational relations, *rational kernels*, that extend kernel methods to the analysis of variable-length sequences or more generally weighted automata. We show that rational kernels can be computed efficiently using a general algorithm of composition of weighted transducers and a general single-source shortest-distance algorithm.

Not all rational kernels are *positive definite and symmetric* (PDS), or equivalently verify the Mercer condition, a condition that guarantees the convergence of training for discriminant classification algorithms such as SVMs. We present several theoretical results related to PDS rational kernels. We show that under some general conditions these kernels are closed under sum, product, or Kleene-closure and give a general method for constructing a PDS rational kernel from an arbitrary transducer defined on some non-idempotent semirings. We give the proof of several characterization results that can be used to guide the design of PDS rational kernels. We also show that some commonly used string kernels or similarity measures such as the edit-distance, the convolution kernels of Haussler, and some string kernels used in the context of computational biology are specific instances of rational kernels. Our results include the proof that the edit-distance over a non-trivial alphabet is not *negative definite*, which, to the best of our knowledge, was never stated or proved before.

Rational kernels can be combined with SVMs to form efficient and powerful techniques for a variety of classification tasks in text and speech processing, or computational biology. We describe examples of general families of PDS rational kernels that are useful in many of these applications and report the result of our experiments illustrating the use of rational kernels in several difficult large-vocabulary spoken-dialog classification tasks based on

deployed spoken-dialog systems. Our results show that rational kernels are easy to design and implement and lead to substantial improvements of the classification accuracy.

1. Introduction

Many classification algorithms were originally designed for fixed-length vectors. Recent applications in text and speech processing and computational biology require however the analysis of variable-length sequences and more generally weighted automata. Indeed, the output of a large-vocabulary speech recognizer for a particular input speech utterance, or that of a complex information extraction system combining several knowledge sources for a specific input query, is typically a weighted automaton compactly representing a large set of alternative sequences. The weights assigned by the system to each sequence are used to rank different alternatives according to the models the system is based on. The error rate of such complex systems is still too high in many tasks to rely only on their one-best output, thus it is preferable instead to use the full weighted automata which contain the correct result in most cases.

An approach widely used in statistical learning techniques such as Support Vector Machines (SVMs) (Boser et al., 1992, Cortes and Vapnik, 1995, Vapnik, 1998) is that of kernel methods, due to their computational efficiency in high-dimensional feature spaces. We introduce a general family of kernels based on weighted transducers or rational relations, *rational kernels*, that extend kernel methods to the analysis of variable-length sequences or more generally weighted automata.[1] We show that rational kernels can be computed efficiently using a general algorithm of composition of weighted transducers and a general single-source shortest-distance algorithm.

Not all rational kernels are *positive definite and symmetric* (PDS), or equivalently verify the Mercer condition (Berg et al., 1984), a condition that guarantees the convergence of training for discriminant classification algorithms such as SVMs. We present several theoretical results related to PDS rational kernels. We show that under some general conditions these kernels are closed under sum, product, or Kleene-closure and give a general method for constructing a PDS rational kernel from an arbitrary transducer defined on some non-idempotent semirings. We give the proof of several characterization results that can be used to guide the design of PDS rational kernels.

We also study the relationship between rational kernels and some commonly used string kernels or similarity measures such as the edit-distance, the convolution kernels of Haussler (Haussler, 1999), and some string kernels used in the context of computational biology (Leslie et al., 2003). We show that these kernels are all specific instances of rational kernels. In each case, we explicitly describe the corresponding weighted transducer. These transducers are often simple and efficient for computing kernels. Their diagram provides more insight into the definition of kernels and can guide the design of new kernels. Our results also include the proof of the fact that the edit-distance over a non-trivial alphabet is not *negative definite*, which, to the best of our knowledge, was never stated or proved before.

Rational kernels can be combined with SVMs to form efficient and powerful techniques for a variety of applications to text and speech processing, or to computational biology. We describe examples of general families of PDS rational kernels that are useful in many of

1. We have described in shorter publications part of the material presented here (Cortes et al., 2003a,b,c,d).

Semiring	Set	$\oplus$	$\otimes$	$\overline{0}$	$\overline{1}$
Boolean	$\{0,1\}$	$\vee$	$\wedge$	0	1
Probability	$\mathbb{R}_+$	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0

Table 1: Semiring examples. $\oplus_{\log}$ is defined by: $x \oplus_{\log} y = -\log(e^{-x} + e^{-y})$.

these applications. We report the result of our experiments illustrating the use of rational kernels in several difficult large-vocabulary spoken-dialog classification tasks based on deployed spoken-dialog systems. Our results show that rational kernels are easy to design and implement and lead to substantial improvements of the classification accuracy.

The paper is organized as follows. In the following section, we introduce the notation and some preliminary algebraic and automata-theoretic definitions used in the remaining sections. Section 3 introduces the definition of rational kernels. In Section 4, we present general algorithms that can be used to compute rational kernels efficiently. Section 5 introduces the classical definitions of positive definite and negative definite kernels and gives a number of novel theoretical results, including the proof of some general closure properties of PDS rational kernels, a general construction of PDS rational kernels starting from an arbitrary weighted transducer, a characterization of acyclic PDS rational kernels, and the proof of the closure properties of a very general class of PDS rational kernels. Section 6 studies the relationship between some commonly used kernels and rational kernels. Finally, the results of our experiments in several spoken-dialog classification tasks are reported in Section 7.

2. Preliminaries

In this section, we present the algebraic definitions and notation needed to introduce rational kernels.

A system $(\mathbb{K}, \odot, e)$ is a *monoid* if it is closed under $\odot$: $a \odot b \in \mathbb{K}$ for all $a, b \in \mathbb{K}$; $\odot$ is associative: $(a \odot b) \odot c = a \odot (b \odot c)$ for all $a, b, c \in \mathbb{K}$; and e is an identity for $\odot$: $a \odot e = e \odot a = a$, for all $a \in \mathbb{K}$. When additionally $\odot$ is commutative: $a \odot b = b \odot a$ for all $a, b \in \mathbb{K}$, then $(\mathbb{K}, \odot, e)$ is said to be a *commutative monoid*.

Definition 1 (Kuich and Salomaa (1986)) *A system* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ *is a* semiring *if:* $(\mathbb{K}, \oplus, \overline{0})$ *is a commutative monoid with identity element* $\overline{0}$*;* $(\mathbb{K}, \otimes, \overline{1})$ *is a monoid with identity element* $\overline{1}$*;* $\otimes$ *distributes over* $\oplus$*; and* $\overline{0}$ *is an annihilator for* $\otimes$*: for all* $a \in \mathbb{K}, a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$.

Thus, a semiring is a ring that may lack negation. Table 1 lists some familiar semirings. In addition to the Boolean semiring and the probability semiring, two semirings often used in applications are the *log semiring* which is isomorphic to the probability semiring via a log morphism, and the *tropical semiring* which is derived from the log semiring using the Viterbi approximation.

Definition 2 *A* weighted finite-state transducer T *over a semiring* $\mathbb{K}$ *is an 8-tuple* $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ *where:* Σ *is the finite input alphabet of the transducer;* Δ *is the finite output alphabet;* Q *is a finite set of states;* $I \subseteq Q$ *the set of initial states;* $F \subseteq Q$ *the set of final states;* $E \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times (\Delta \cup \{\epsilon\}) \times \mathbb{K} \times Q$ *a finite set of transitions;* $\lambda : I \to \mathbb{K}$ *the initial weight function; and* $\rho : F \to \mathbb{K}$ *the final weight function mapping* F *to* $\mathbb{K}$.

Weighted automata can be formally defined in a similar way by simply omitting the input or output labels.

Given a transition $e \in E$, we denote by $p[e]$ its origin or previous state and $n[e]$ its destination state or next state, and $w[e]$ its weight. A *path* $\pi = e_1 \cdots e_k$ 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* π is a path whose origin and destination coincide: $p[\pi] = n[\pi]$. A weighted automaton or transducer is said to be *acyclic* if it admits no cycle. A *successful path* in a weighted automaton or transducer M is a path from an initial state to a final state. The weight function w can also be extended to paths by defining the weight of a path as the $\otimes$-product of the weights of its constituent transitions: $w[\pi] = w[e_1] \otimes \cdots \otimes w[e_k]$. We denote by $P(q, q')$ the set of paths from q to q' and by $P(q, x, y, q')$ the set of paths from q to q' with input label $x \in \Sigma^*$ and output label $y \in \Delta^*$. These definitions can be extended to subsets $R, R' \subseteq Q$, by: $P(R, x, y, R') = \cup_{q \in R,\, q' \in R'} P(q, x, y, q')$. We denote by $w[M]$ the $\oplus$-sum of the weights of all the successful paths of the automaton or transducer M, when that sum is well-defined and in $\mathbb{K}$. A transducer T is *regulated* if the output weight associated by T to any pair of input-output string (x, y) by:

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

is well-defined and in $\mathbb{K}$. $[\![T]\!](x, y) = \overline{0}$ when $P(I, x, y, F) = \emptyset$. If for all $q \in Q$, the sum $\bigoplus_{\pi \in P(q, \epsilon, \epsilon, q)} w[\pi]$ is in $\mathbb{K}$, then T is regulated. In particular, when T does not have any ϵ-cycle, that is a cycle labeled with ϵ (both input and output labels), it is regulated. In the following, we will assume that all the transducers considered are regulated. Regulated weighted transducers are closed under the rational operations: $\oplus$-sum, $\otimes$-product and Kleene-closure which are defined as follows for all transducers T_1 and T_2 and $(x, y) \in \Sigma^* \times \Delta^*$:

$$[\![T_1 \oplus T_2]\!](x, y) = [\![T_1]\!](x, y) \oplus [\![T_2]\!](x, y) \tag{2}$$

$$[\![T_1 \otimes T_2]\!](x, y) = \bigoplus_{x = x_1 x_2, y = y_1 y_2} [\![T_1]\!](x_1, y_1) \otimes [\![T_2]\!](x_2, y_2) \tag{3}$$

$$[\![T^*]\!](x, y) = \bigoplus_{n=0}^{\infty} T^n(x, y) \tag{4}$$

where T^n stands for the $(n-1)$-$\otimes$-product of T with itself.

For any transducer T, we denote by T^{-1} its *inverse*, that is the transducer obtained from T by transposing the input and output labels of each transition and the input and output alphabets.

Composition is a fundamental operation on weighted transducers that can be used in many applications to create complex weighted transducers from simpler ones. Let

$T_1 = (\Sigma, \Delta, Q_1, I_1, F_1, E_1, \lambda_1, \rho_1)$ and $T_2 = (\Delta, \Omega, Q_2, I_2, F_2, E_2, \lambda_2, \rho_2)$ be two weighted transducers defined over a commutative semiring $\mathbb{K}$ such that Δ, the output alphabet of T_1, coincides with the input alphabet of T_2. Then, the result of the composition of T_1 and T_2 is a weighted transducer $T_1 \circ T_2$ which, when it is regulated, is defined for all x, y by (Berstel, 1979, Eilenberg, 1974, Salomaa and Soittola, 1978, Kuich and Salomaa, 1986):[2]

$$[\![T_1 \circ T_2]\!](x, y) = \bigoplus_{z \in \Delta^*} [\![T_1]\!](x, z) \otimes [\![T_2]\!](z, y) \tag{5}$$

Note that a transducer can be viewed as a matrix over a countable set $\Sigma^* \times \Delta^*$ and composition as the corresponding matrix-multiplication.

The definition of composition extends naturally to weighted automata since a weighted automaton can be viewed as a weighted transducer with identical input and output labels for each transition. The corresponding transducer associates $[\![A]\!](x)$ to a pair (x, x), and 0 to all other pairs. Thus, the composition of a weighted automaton $A_1 = (\Delta, Q_1, I_1, F_1, E_1, \lambda_1, \rho_1)$ and a weighted transducer $T_2 = (\Delta, \Omega, Q_2, I_2, F_2, E_2, \lambda_2, \rho_2)$ is simply defined for all x, y in $\Delta^* \times \Omega^*$ by:

$$[\![A_1 \circ T_2]\!](x, y) = \bigoplus_{x \in \Delta^*} [\![A_1]\!](x) \otimes [\![T_2]\!](x, y) \tag{6}$$

when these sums are well-defined and in $\mathbb{K}$. *Intersection* of two weighted automata is the special case of composition where both operands are weighted automata, or equivalently weighted transducers with identical input and output labels for each transition.

3. Definitions

Let X and Y be non-empty sets. A function $K : X \times Y \to \mathbb{R}$ is said to be a *kernel* over $X \times Y$. This section introduces *rational kernels*, which are kernels defined over sets of strings or weighted automata.

Definition 3 *A kernel K over $\Sigma^* \times \Delta^*$ is said to be* rational *if there exist a weighted transducer $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ over the semiring $\mathbb{K}$ and a function $\psi : \mathbb{K} \to \mathbb{R}$ such that for all $x \in \Sigma^*$ and $y \in \Delta^*$:*[3]

$$K(x, y) = \psi([\![T]\!](x, y)) \tag{7}$$

K is then said to be defined by the pair (ψ, T).

This definition and many of the results presented in this paper can be generalized by replacing the free monoids Σ^* and Δ^* with arbitrary monoids M_1 and M_2. Also, note that we are not making any particular assumption about the function ψ in this definition. In general, it is an arbitrary function mapping $\mathbb{K}$ to $\mathbb{R}$.

Figure 1 shows an example of a weighted transducer over the probability semiring corresponding to the gappy n-gram kernel with decay factor λ as defined by (Lodhi et al., 2001). Such gappy n-gram kernels are rational kernels (Cortes et al., 2003c).

2. We use a *matrix notation* for the definition of composition as opposed to a *functional notation*.
3. We chose to call these kernels "rational" because their definition is based on *rational relations* or *rational transductions* (Salomaa and Soittola, 1978, Kuich and Salomaa, 1986) represented by a weighted transducer. The mathematical counterpart of weighted automata and transducers are also called *rational power series* Berstel and Reutenauer (1988) which further justifies our terminology.

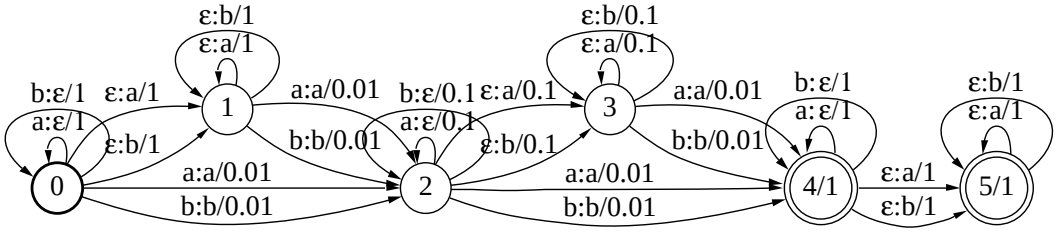


Figure 1: Gappy bigram rational kernel with decay factor $\lambda = .1$. Bold face circles represent initial states and double circles indicate final states. Inside each circle, the first number indicates the state number, the second, at final states only, the value of the final weight function ρ at that state. Arrows represent transitions. They are labeled with an input and an output symbol separated by a colon and followed by their corresponding weight after the slash symbol.

Rational kernels can be naturally extended to kernels over weighted automata. Let A be a weighted automaton defined over the semiring $\mathbb{K}$ and the alphabet Σ and B a weighted automaton defined over the semiring $\mathbb{K}$ and the alphabet Δ, $K(A, B)$ is defined by:

$$K(A, B) = \psi \left(\bigoplus_{(x,y) \in \Sigma^* \times \Delta^*} [\![A]\!](x) \otimes [\![T]\!](x, y) \otimes [\![B]\!](y) \right) \tag{8}$$

for all weighted automata A and B such that the $\oplus$-sum:

$$\bigoplus_{(x,y) \in \Sigma^* \times \Delta^*} [\![A]\!](x) \otimes [\![T]\!](x, y) \otimes [\![B]\!](y)$$

is well-defined and in $\mathbb{K}$. This sum is always defined and in $\mathbb{K}$ when A and B are acyclic weighted automata since the sum then runs over a finite set. It is defined for all weighted automata in all *closed semirings* (Kuich and Salomaa, 1986) such as the tropical semiring. In the probability semiring, the sum is well-defined for all A, B, and T representing probability distributions. When $K(A, B)$ is defined, Equation 8 can be equivalently written as:

$$K(A, B) = \psi \left(\bigoplus_{(x,y) \in \Sigma^* \times \Delta^*} [\![A \circ T \circ B]\!](x, y) \right) = \psi(w[A \circ T \circ B]) \tag{9}$$

The next section presents a general algorithm for computing rational kernels.

4. Algorithms

The algorithm for computing $K(x, y)$, or $K(A, B)$, for any two acyclic weighted automata, or for any two weighted automata such that the sum above is well-defined, is based on two general algorithms that we briefly present: composition of weighted transducers to combine A, T, and B, and a general shortest-distance algorithm in a semiring $\mathbb{K}$ to compute the $\oplus$-sum of the weights of all successful paths of the composed transducer.

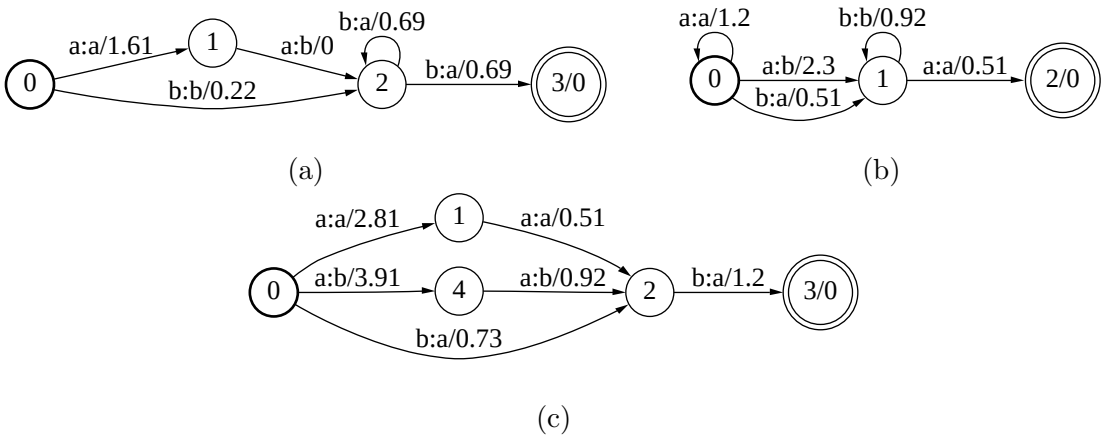


Figure 2: (a) Weighted transducer T_1 over the log semiring. (b) Weighted transducer T_2 over the log semiring. (c) $T_1 \circ T_2$, result of the composition of T_1 and T_2.

4.1 Composition of weighted transducers

There exists a general and efficient composition algorithm for weighted transducers which takes advantage of the sparseness of the input transducers (Pereira and Riley, 1997, Mohri et al., 1996). States in the composition $T_1 \circ T_2$ of two weighted transducers T_1 and T_2 are identified with pairs of a state of T_1 and a state of T_2. Leaving aside transitions with ϵ inputs or outputs, the following rule specifies how to compute a transition of $T_1 \circ T_2$ from appropriate transitions of T_1 and T_2:[4]

$$(q_1, a, b, w_1, q_2) \quad \text{and} \quad (q_1', b, c, w_2, q_2') \Longrightarrow ((q_1, q_1'), a, c, w_1 \otimes w_2, (q_2, q_2')) \tag{10}$$

In the worst case, all transitions of T_1 leaving a state q_1 match all those of T_2 leaving state q_1', thus the space and time complexity of composition is quadratic: $O((|Q_1|+|E_1|)(|Q_2|+|E_2|))$. Figure 2 illustrates the algorithm when applied to the transducers of Figure 2 (a)-(b) defined over the log semiring.

4.2 Single-source shortest distance algorithm over a semiring

Given a weighted automaton or transducer M, the *shortest-distance* from state q to the set of final states F is defined as the $\oplus$-sum of all the paths from q to F:

$$d[q] = \bigoplus_{\pi \in P(q,F)} w[\pi] \otimes \rho[n[\pi]] \tag{11}$$

when this sum is well-defined and in $\mathbb{K}$. This is always the case when the semiring is *k-closed* or when M is acyclic (Mohri, 2002), the case of interest in our experiments. There exists a general algorithm for computing the shortest-distance $d[q]$ (Mohri, 2002). The algorithm is based on a generalization to k-closed semirings of the relaxation technique used in classical

4. See (Pereira and Riley, 1997, Mohri et al., 1996) for a detailed presentation of the algorithm including the use of a transducer filter for dealing with ϵ-multiplicity in the case of non-idempotent semirings.

single-source shortest-paths algorithms. When M is acyclic, the complexity of the algorithm is linear: $O(|Q|+(T_\oplus+T_\otimes)|E|)$, where $T_\oplus$ denotes the maximum time to compute $\oplus$ and $T_\otimes$ the time to compute $\otimes$ (Mohri, 2002). The algorithm can then be viewed as a generalization of Lawler's algorithm (Lawler, 1976) to the case of an arbitrary semiring $\mathbb{K}$. It is then based on a generalized relaxation of the outgoing transitions of each state of M visited in reverse topological order (Mohri, 2002).

Let K be a rational kernel and let T be the associated weighted transducer. Let A and B be two acyclic weighted automata or, more generally, two weighted automata such that the sum in Equation 9 is well-defined and in $\mathbb{K}$. A and B may represent just two strings $x, y \in \Sigma^*$ or may be any complex weighted automata. By definition of rational kernels (Equation 9) and the shortest-distance (Equation 11), $K(A, B)$ can be computed by:

1. Constructing the composed transducer $N = A \circ T \circ B$.
2. Computing $w[N]$, by determining the shortest-distance from the initial states of N to its final states using the shortest-distance algorithm just described.
3. Computing $\psi(w[N])$.

When A and B are acyclic, the shortest-distance algorithm is linear and the total complexity of the algorithm is $O(|T||A||B|+\Phi)$, where $|T|$, $|A|$, and $|B|$ denote respectively the size of T, A and B and Φ the worst case complexity of computing $\psi(x)$, $x \in \mathbb{K}$. If we assume that Φ can be computed in constant time as in many applications, then the complexity of the computation of $K(A, B)$ is quadratic with respect to A and B: $O(|T||A||B|)$.

5. Theory of PDS and NDS Rational Kernels

In learning techniques such as those based on SVMs, we are particularly interested in kernels that are *positive definite symmetric* (PDS), or, equivalently, kernels verifying Mercer's condition, which guarantee the existence of a Hilbert space and a dot product associated to the kernel considered. This ensures the convergence of the training algorithm to a unique optimum. Thus, in what follows, we will focus on theoretical results related to the construction of rational kernels that are PDS. Due to the symmetry condition, the input and output alphabets Σ and Δ will coincide for the underlying transducers associated to the kernels.

This section reviews a number of results related to general PDS kernels, that is the class of all kernels that have the Mercer property (Berg et al., 1984). It also gives novel proofs and results in the specific case of *PDS rational kernels*. These results can be used to combine PDS rational kernels to design new PDS rational kernels or to construct a PDS rational kernel. Our proofs and results are original and are not just straightforward extensions of those existing in the case of general PDS kernels. This is because, for example, a closure property for PDS rational kernels must guarantee not just that the PDS property is preserved but also that the rational property is retained. Our original results include a general construction of PDS rational kernels from arbitrary weighted transducers, a number of theorems related to the converse, and a study of the *negative definiteness* of some rational kernels.

Definition 4 *Let X be a non-empty set. A function $K : X \times X \to \mathbb{R}$ is said to be a PDS kernel if it is symmetric ($K(x, y) = K(y, x)$ for all $x, y \in X$) and*

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \tag{12}$$

for all $n \geq 1$, $\{x_1, \ldots, x_n\} \subseteq X$ and $\{c_1, \ldots, c_n\} \subseteq \mathbb{R}$.

It is clear from classical results of linear algebra that K is a PDS kernel iff the matrix $K(x_i, x_j)_{i,j \leq n}$ for all $n \geq 1$ and all $\{x_1, \ldots, x_n\} \subseteq X$ is symmetric and all its eigenvalues are non-negative.

PDS kernels can be used to construct other families of kernels that also meet these conditions (Schölkopf and Smola, 2002). *Polynomial kernels* of degree p are formed from the expression $(K+a)^p$, and *Gaussian kernels* can be formed as $\exp(-d^2/\sigma^2)$ with $d^2(x, y) = K(x, x) + K(y, y) - 2K(x, y)$. The following sections will provide other ways of constructing PDS rational kernels.

5.1 General Closure Properties of PDS Kernels

The following theorem summarizes general closure properties of PDS kernels (Berg et al., 1984).

Theorem 5 *Let X and Y be two non-empty sets.*

1. Closure under sum*: Let $K_1, K_2 : X \times X \to \mathbb{R}$ be PDS kernels, then $K_1 + K_2 : X \times X \to \mathbb{R}$ is a PDS kernel.*
2. Closure under product*: Let $K_1, K_2 : X \times X \to \mathbb{R}$ be PDS kernels, then $K_1 \cdot K_2 : X \times X \to \mathbb{R}$ is a PDS kernel.*
3. Closure under tensor product*: Let $K_1 : X \times X \to \mathbb{R}$ and $K_2 : Y \times Y \to \mathbb{R}$ be PDS kernels, then their tensor product $K_1 \odot K_2 : (X \times Y) \times (X \times Y) \to \mathbb{R}$, where $K_1 \odot K_2((x_1, y_1), (x_2, y_2)) = K_1(x_1, x_2) \cdot K_2(y_1, y_2)$ is a PDS kernel.*
4. Closure under pointwise limit*: Let $K_n : X \times X \to \mathbb{R}$ be a PDS kernel for all $n \in \mathbb{N}$ and assume that $\lim_{n \to \infty} K_n(x, y)$ exists for all $x, y \in X$, then K defined by $K(x, y) = \lim_{n \to \infty} K_n(x, y)$ is a PDS kernel.*
5. Closure under composition with a power series*: Let $K : X \times X \to \mathbb{R}$ be a PDS kernel such that $|K(x, y)| < \rho$ for all $(x, y) \in X \times X$. Then if the radius of convergence of the power series $S = \sum_{n=0}^{\infty} a_n x^n$ is ρ and $a_n \geq 0$ for all $n \geq 0$, the composed kernel $S \circ K$ is a PDS kernel. In particular, if $K : X \times X \to \mathbb{R}$ is a PDS kernel, then so is $\exp(K)$.*

In particular, these closure properties all apply to PDS kernels that are rational, e.g., the sum or product of two PDS rational kernels is a PDS kernel. However, Theorem 5 does not guarantee the result to be a rational kernel. In the next section, we examine precisely the question of the closure properties of PDS rational kernels (under rational operations).

5.2 Closure Properties of PDS Rational Kernels

In this section, we assume that a fixed function ψ is used in the definition of all the rational kernels mentioned. We denote by K_T the rational kernel corresponding to the transducer T and defined for all $x, y \in \Sigma^*$ by $K_T(x, y) = \psi([\![T]\!](x, y))$.

Theorem 6 *Let Σ be a non-empty alphabet. The following closure properties hold for PDS rational kernels.*

1. Closure under $\oplus$-sum*: Assume that $\psi : (\mathbb{K}, \oplus, \overline{0}) \to (\mathbb{R}, +, 0)$ is a monoid morphism.*[5] *Let $K_{T_1}, K_{T_2} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ be PDS rational kernels, then $K_{T_1 \oplus T_2} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ is a PDS rational kernel and $K_{T_1 \oplus T_2} = K_{T_1} + K_{T_2}$.*

2. Closure under $\otimes$-product*: Assume that $\psi : (\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1}) \to (\mathbb{R}, +, \times, 0, 1)$ is a semiring morphism. Let $K_{T_1}, K_{T_2} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ be PDS rational kernels, then $K_{T_1 \otimes T_2} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ is a PDS rational kernel.*

3. Closure under Kleene-closure*: Assume that $\psi : (\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1}) \to (\mathbb{R}, +, \times, 0, 1)$ is a continuous semiring morphism. Let $K_T : \Sigma^* \times \Sigma^* \to \mathbb{R}$ be a PDS rational kernel, then $K_{T^*} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ is a PDS rational kernel.*

Proof The closure under $\oplus$-sum follows directly from Theorem 5 and the fact that for all $x, y \in \Sigma^*$:

$$\psi([\![T_1]\!](x, y) \oplus [\![T_2]\!](x, y)) = \psi([\![T_1]\!](x, y)) + \psi([\![T_2]\!](x, y))$$

when $\psi : (\mathbb{K}, \oplus, \overline{0}) \to (\mathbb{R}, +, 0)$ is a monoid morphism. For the closure under $\otimes$-product, when ψ is a semiring morphism, for all $x, y \in \Sigma^*$:

$$\begin{aligned} \psi([\![T_1 \otimes T_2]\!](x, y)) &= \sum_{x_1 x_2 = x, y_1 y_2 = y} \psi([\![T_1]\!](x_1, y_1)) \cdot \psi([\![T_2]\!](x_2, y_2)) \\ &= \sum_{x_1 x_2 = x, y_1 y_2 = y} K_{T_1} \odot K_{T_2}((x_1, x_2), (y_1, y_2)) \end{aligned} \tag{13}$$

By Theorem 5, since K_{T_1} and K_{T_2} are PDS kernels, their tensor product $K_{T_1} \odot K_{T_2}$ is a PDS kernel and there exists a Hilbert space $H \subseteq \mathbb{R}^{\Sigma^*}$ and a mapping $u \to \phi_u$ such that $K_{T_1} \odot K_{T_2}(u, v) = \langle \phi_u, \phi_v \rangle$ (Berg et al., 1984). Thus

$$\begin{aligned} \psi([\![T_1 \otimes T_2]\!](x, y)) &= \sum_{x_1 x_2 = x, y_1 y_2 = y} \langle \phi_{(x_1, x_2)}, \phi_{(y_1, y_2)} \rangle \\ &= \left\langle \sum_{x_1 x_2 = x} \phi_{(x_1, x_2)}, \sum_{y_1 y_2 = y} \phi_{(y_1, y_2)} \right\rangle \end{aligned} \tag{14}$$

Since a dot product is positive definite, this equality implies that $K_{T_1 \otimes T_2}$ is a PDS kernel. A similar proof is given by Haussler (1999). The closure under Kleene-closure is a direct

5. A *monoid morphism* $\psi : (\mathbb{K}, \oplus, \overline{0}) \to (\mathbb{R}, +, 0)$ is a function verifying $\psi(x \oplus y) = \psi(x) + \psi(y)$ for all $x, y \in \mathbb{K}$, and $\psi(\overline{0}) = 0$. A *semiring morphism* ψ is a function $\psi : (\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1}) \to (\mathbb{R}, +, \times, 0, 1)$ further verifying $\psi(x \otimes y) = \psi(x) \cdot \psi(y)$ for all $x, y \in \mathbb{K}$, and $\psi(\overline{1}) = 1$.

consequence of the closure under $\oplus$-sum and $\otimes$-product of PDS rational kernels and the closure under pointwise limit of PDS kernels (Theorem 5). ■

Theorem 6 provides a general method for constructing complex PDS rational kernels from simpler ones. PDS rational kernels defined to model specific prior knowledge sources can be combined using rational operations to create a more general PDS kernel.

In contrast to Theorem 6, PDS rational kernels are not closed under composition. This is clear since the ordinary matrix multiplication does not preserve positive definiteness in general.

The next section studies a general construction of PDS rational kernels using composition.

5.3 A General Construction of PDS Rational Kernels

In this section, we assume that $\psi : (\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1}) \to (\mathbb{R}, +, \times, 0, 1)$ is a continuous semiring morphism. This limits the choice of the semiring associated to the weighted transducer defining a rational kernel, since it needs in particular to be commutative and non-idempotent.[6] Our study of PDS rational kernels in this section is thereby limited to such semirings. This should not leave the reader with the incorrect perception that all PDS rational kernels are defined over non-idempotent semirings though. As already mentioned before, in general, the function ψ can be chosen arbitrarily and needs not impose any algebraic property on the semiring used.

We show that there exists a general way of constructing a PDS rational kernel from any weighted transducer T. The construction is based on composing T with its inverse T^{-1}.

Proposition 7 *Let $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ be a weighted finite-state transducer defined over the semiring $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$. Assume that the weighted transducer $T \circ T^{-1}$ is regulated, then $(\psi, T \circ T^{-1})$ defines a PDS rational kernel over $\Sigma^* \times \Sigma^*$.*

Proof Denote by S the composed transducer $T \circ T^{-1}$. Let K be the rational kernel defined by S. By definition of composition

$$K(x, y) = \psi([\![S]\!](x, y)) = \psi\left(\bigoplus_{z \in \Delta^*} [\![T]\!](x, z) \otimes [\![T]\!](y, z)\right) \tag{15}$$

for all $x, y \in \Sigma^*$. Since ψ is a continuous semiring morphism, for all $x, y \in \Sigma^*$

$$K(x, y) = \psi([\![S]\!](x, y)) = \sum_{z \in \Delta^*} \psi([\![T]\!](x, z)) \cdot \psi([\![T]\!](y, z)) \tag{16}$$

For all $n \in \mathbb{N}$ and $x, y \in \Sigma^*$, define $K_n(x, y)$ by:

$$K_n(x, y) = \sum_{|z| \leq n} \psi([\![T]\!](x, z)) \cdot \psi([\![T]\!](y, z)) \tag{17}$$

6. If $\mathbb{K}$ is idempotent, for any $x \in \mathbb{K}$, $\psi(x) = \psi(x \oplus x) = \psi(x) + \psi(x) = 2\psi(x)$, which implies that $\psi(x) = 0$ for all x.

where the sum runs over all strings $z \in \Delta^*$ of length less than or equal to n. Clearly, K_n defines a symmetric kernel. For any $l \geq 1$ and any $x_1, \ldots, x_l \in \Sigma^*$, define the matrix M_n by: $M_n = (K_n(x_i, x_j))_{i \leq l, j \leq l}$. Let $z_1, z_2, \ldots, z_m$ be an arbitrary ordering of the strings of length less than or equal to n. Define the matrix A by:

$$A = (\psi([\![T]\!](x_i, z_j)))_{i \leq l; j \leq m} \tag{18}$$

By definition of K_n, $M_n = AA^t$. The eigenvalues of AA^t are non-negative for any rectangular matrix A, thus K_n is a PDS kernel. Since K is a pointwise limit of K_n, $K(x, y) = \lim_{n \to \infty} K_n(x, y)$, by Theorem 5, K is a PDS kernel. This ends the proof of the proposition. ■

The next propositions provide results related to the converse of Proposition 7. We denote by $Id_{\mathbb{R}}$ the identity function over $\mathbb{R}$.

Proposition 8 *Let $S = (\Sigma, \Sigma, Q, I, F, E, \lambda, \rho)$ be an acyclic weighted finite-state transducer over $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ such that (ψ, S) defines a PDS rational kernel on $\Sigma^* \times \Sigma^*$, then there exists a weighted transducer T over the probability semiring such that $(Id_{\mathbb{R}}, T \circ T^{-1})$ defines the same rational kernel.*

Proof Let S be an acyclic weighted transducer verifying the hypotheses of the proposition. Let $X \subset \Sigma^*$ be the finite set of strings accepted by S. Since S is symmetric, $X \times X$ is the set of pairs of strings (x, y) defining the rational relation associated with S. Let $x_1, x_2, \ldots, x_n$ be an arbitrary numbering of the elements of X. Define the matrix M by:

$$M = (\psi([\![S]\!](x_i, x_j)))_{1 \leq i \leq n, 1 \leq j \leq n} \tag{19}$$

Since S defines a PDS rational kernel, M is a symmetric matrix with non-negative eigenvalues, i.e., M is symmetric positive semi-definite. The Cholesky decomposition extends to the case of semi-definite matrices (Dongarra et al., 1979): there exists an upper triangular matrix $R = (R_{ij})$ with non-negative diagonal elements such that $M = RR^t$. Let $Y = \{y_1, \ldots, y_n\}$ be an arbitrary subset of n distinct strings of Σ^*. Define the weighted transducer T over the $X \times Y$ by:

$$[\![T]\!](x_i, y_j) = R_{ij} \tag{20}$$

for all i, j, $1 \leq i, j \leq n$. By definition of composition, $[\![T \circ T^{-1}]\!](x_i, x_j) = \psi([\![S]\!](x_i, x_j))$ for all i, j, $1 \leq i, j \leq n$. Thus, $T \circ T^{-1} = \psi(S)$, which proves the claim of the proposition. ■

Note that when the matrix M introduced in the proof is positive definite, that is when the eigenvalues of the matrix associated with S are all positive, then Cholesky's decomposition and thus the weights associated to the input strings of T are unique.

Assume that the same continuous semiring morphism ψ is used in the definition of all the rational kernels.

Proposition 9 *Let Θ be the subset of the weighted transducers over $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ such that for any $S \in \Theta$, (ψ, S) defines a PDS rational kernel and there exists a weighted transducer $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ over the probability semiring such that $(Id_{\mathbb{R}}, T \circ T^{-1})$ defines the same rational kernel as (ψ, S). Then Θ is closed under $\oplus$-sum, $\otimes$-product, and Kleene-closure.*

Proof Let $S_1, S_2 \in \Theta$, we will show that $S_1 \oplus S_2 \in \Theta$, $S_1 \otimes S_2 \in \Theta$, and $S_1^* \in \Theta$. By definition, there exist $T_1 = (\Sigma, \Delta_1, Q_1, I_1, F_1, E_1, \lambda_1, \rho_1)$ and $T_2 = (\Sigma, \Delta_2, Q_2, I_2, F_2, E_2, \lambda_2, \rho_2)$ such that:

$$K_1 = T_1 \circ T_1^{-1} \quad \text{and} \quad K_2 = T_2 \circ T_2^{-1} \tag{21}$$

where K_1 (K_2) is the PDS rational kernel defined by (ψ, S_1) (resp. (ψ, S_2)). Let u be an alphabetic morphism mapping Δ_2 to a new alphabet Δ_2' such that $\Delta_1 \cap \Delta_2' = \emptyset$. u is clearly a rational transduction (Berstel, 1979) and can be represented by a finite-state transducer U. Thus, we can define a new weighted transducer T_2' by: $T_2' = T_2 \circ U = (\Sigma, \Delta_2', Q_2, I_2, F_2, E_2', \lambda_2, \rho_2)$, which only differs from T_2 by some renaming of its output labels. This does not affect the result of the composition with the inverse transducer since $U \circ U^{-1}$ is the identity mapping over Δ_2^*:

$$T_2' \circ {T'}_2^{-1} = T_2 \circ U \circ (U^{-1} \circ T_2^{-1}) = T_2 \circ T_2^{-1} = K_2 \tag{22}$$

Since, T_1 and T_2 have distinct output alphabets, their output labels cannot match, thus:

$$T_1 \circ {T'}_2^{-1} = \emptyset \quad \text{and} \quad T_2' \circ T_1^{-1} = \emptyset \tag{23}$$

Let $T = T_1 + T_2'$, in view of Equation 22 and Equation 23:

$$T \circ T^{-1} = (T_1 + T_2') \circ (T_1 + T_2')^{-1} = (T_1 \circ T_1^{-1}) + ({T'}_2 \circ {T'}_2^{-1}) = K_1 + K_2 \tag{24}$$

Since the same continuous semiring morphism ψ is used for the definition of all the rational kernels in Θ, by Theorem 6, $K_1 + K_2$ is a PDS rational kernel defined by $(\psi, S_1 \oplus S_2)$ and $S_1 \oplus S_2$ is in Θ. Similarly, define T' as $T' = T_1 \cdot T_2'$.

$$T' \circ T'^{-1} = (T_1 \cdot T_2') \circ (T_1 \cdot T_2')^{-1} = (T_1 \circ T_1^{-1}) \cdot (T_2 \circ {T'}_2^{-1}) \tag{25}$$

Thus, $S_1 \otimes S_2$ is in Θ. Let x be a symbol not in Δ_1 and let $\Delta_1' = \Delta_1 \cup \{x\}$. Let V be the finite-state transducer accepting as input only ϵ and mapping ϵ to x and define T_1' by $T_1' = V \cdot T_1$. Since x does not match any of the output labels of T_1, $T_1' \circ {T_1'}^{-1} = T_1 \circ {T_1}^{-1}$ and $(T_1' \circ {T_1'}^{-1})^* = {T_1'}^* \circ ({T_1'}^{-1})^*$:

$$(T_1 \circ {T_1}^{-1})^* = (T_1' \circ {T_1'}^{-1})^* = {T_1'}^* \circ ({T_1'}^{-1})^* \tag{26}$$

Thus, by Theorem 6, S_1^* is a PDS rational kernel that is in Θ. ■

Proposition 9 raises the following question: under the same assumptions, are all PDS rational kernels defined by a pair of the form $(\psi, T \circ T^{-1})$? A natural conjecture is that this is the case and that this property provides a characterization of the weighted transducers defining PDS rational kernels. Propositions 8 and 9 both favor that conjecture. Proposition 8 shows that this holds in the acyclic case. Proposition 9 might be useful to extend this to the general case.

In the case of PDS rational kernels defined by $(Id_{\mathbb{R}}, S)$ with S a weighted transducer over the probability semiring, the conjecture could be reformulated as: is S of the form $S = T \circ T^{-1}$? If true, this could be viewed as a generalization of Cholesky's decomposition theorem to the case of infinite matrices given by weighted transducers over the probability semiring.

This ends our discussion of PDS rational kernels. In the next section, we will examine *negative definite kernels* and their relationship with PDS rational kernels.

5.4 Negative Definite Kernels

As mentioned before, given a set X and a distance or dissimilarity measure $d : X \times X \to \mathbb{R}_+$, a common method used to define a kernel K is the following. For all $x, y \in X$,

$$K(x, y) = \exp(-td^2(x, y)) \tag{27}$$

where $t > 0$ is some constant typically used for normalization. Gaussian kernels are defined in this way. However, such kernels K are not necessarily positive definite, e.g., for $X = \mathbb{R}$, $d(x, y) = |x - y|^p$, $p > 1$ and $t = 1$, K is not positive definite. The positive definiteness of K depends on t and the properties of the function d. The classical results presented in this section exactly address such questions (Berg et al., 1984). They include a characterization of PDS kernels based on *negative definite kernels* which may be viewed as distances with some specific properties.[7]

The results we are presenting are general, but we are particularly interested in the case where d can be represented by a rational kernel. We will use these results later when dealing with the case of the edit-distance.

Definition 10 *Let X be a non-empty set. A function $K : X \times X \to \mathbb{R}$ is said to be a* negative definite symmetric kernel *(NDS kernel) if it is symmetric ($K(x, y) = K(y, x)$ for all $x, y \in X$) and*

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \leq 0 \tag{28}$$

for all $n \geq 1$, $\{x_1, \ldots, x_n\} \subseteq X$ and $\{c_1, \ldots, c_n\} \subseteq \mathbb{R}$ with $\sum_{i=1}^{n} c_i = 0$.

Clearly, if K is a PDS kernel then $-K$ is a NDS kernel, however the converse does not hold in general. Negative definite kernels often correspond to distances, e.g., $K(x, y) = (x - y)^\alpha$, $x, y \in \mathbb{R}$, with $0 < \alpha \leq 2$ is a negative definite kernel.

The next theorem summarizes general closure properties of NDS kernels (Berg et al., 1984).

Theorem 11 *Let X be a non-empty set.*

1. Closure under sum*: Let $K_1, K_2 : X \times X \to \mathbb{R}$ be NDS kernels, then $K_1 + K_2 : X \times X \to \mathbb{R}$ is a NDS kernel.*

2. Closure under log and exponentiation*: Let $K : X \times X \to \mathbb{R}$ be a NDS kernel with $K \geq 0$, and α a real number with $0 < \alpha < 1$, then $\log(1 + K), K^\alpha : X \times X \to \mathbb{R}$ are NDS kernels.*

3. Closure under pointwise limit*: Let $K_n : X \times X \to \mathbb{R}$ be a NDS kernel for all $n \in \mathbb{N}$, then K defined by $K(x, y) = \lim_{n \to \infty} K_n(x, y)$ is a NDS kernel.*

7. Many of the results given by Berg et al. (1984) are re-presented in (Schölkopf, 2001) with the terminology of *conditionally positive definite* instead of *negative definite kernels*. We adopt the original terminology used by Berg et al. (1984).

The following theorem clarifies the relation between NDS and PDS kernels and provides in particular a way of constructing PDS kernels from NDS ones (Berg et al., 1984).

Theorem 12 *Let X be a non-empty set, $x_o \in X$, and let $K : X \times X \to \mathbb{R}$ be a symmetric kernel.*

1. *K is negative definite iff $\exp(-tK)$ is positive definite for all $t > 0$.*
2. *Let K' be the function defined by:*

$$K'(x, y) = K(x, x_0) + K(y, x_0) - K(x, y) - K(x_0, x_0) \tag{29}$$

Then K is negative definite iff K' is positive definite.

The theorem gives two ways of constructing a positive definite kernel using a negative definite kernel. The first construction is similar to the way Gaussian kernels are defined. The second construction has been put forward by (Schölkopf, 2001).

6. Relationship with some commonly used kernels or similarity measures

This section studies the relationships between several families of kernels or similarities measures and rational kernels.

6.1 Edit-Distance

A common similarity measure in many applications is that of the *edit-distance*, that is the minimal cost of a series of edit operations (symbol insertions, deletions, or substitutions) transforming one string into the other (Levenshtein, 1966). We denote by $d_e(x, y)$ the edit-distance between two strings x and y over the alphabet Σ with cost 1 assigned to all edit operations.

Proposition 13 *Let Σ be a non-empty finite alphabet and let d_e be the edit-distance over Σ, then d_e is a symmetric rational kernel. Furthermore, (1): d_e is not a PDS kernel, and (2): d_e is a NDS kernel iff $|\Sigma| = 1$.*

Proof The edit-distance between two strings, or weighted automata, can be represented by a simple weighted transducer over the tropical semiring (Mohri, 2003). Since the edit-distance is symmetric, d_e is a symmetric rational kernel. Figure 3(a) shows the corresponding transducer when the alphabet is $\Sigma = \{a, b\}$. The cost of the alignment between two sequences can also be computed by a weighted transducer over the probability semiring (Mohri, 2003), see Figure 3(b).

Let $a \in \Sigma$, then the matrix $(d_e(x_i, x_j))_{1 \leq i,j \leq 2}$ with $x_1 = \epsilon$ and $x_2 = a$ has a negative eigenvalue (-1), thus d_e is not a PDS kernel.

When $|\Sigma| = 1$, the edit-distance simply measures the absolute value of the difference of length between two strings. A string $x \in \Sigma^*$ can then be viewed as a vector of the Hilbert space $\mathbb{R}^\infty$. Denote by $\|\cdot\|$ the corresponding norm. For all $x, y \in \Sigma^*$:

$$d_e(x, y) = \|x - y\|$$

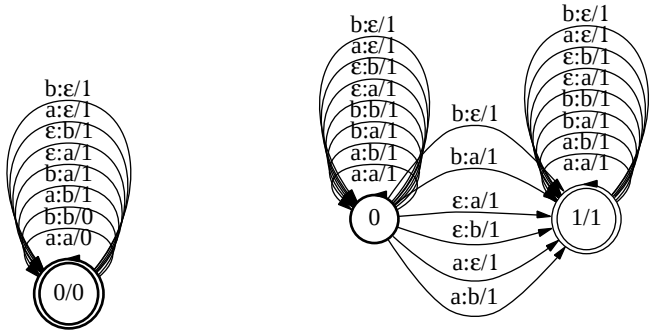


Figure 3: (a) Weighted transducer over the tropical semiring representing the edit-distance over the alphabet $\Sigma = \{a, b\}$. (b) Weighted transducer over the probability semiring computing the cost of alignments over the alphabet $\Sigma = \{a, b\}$.

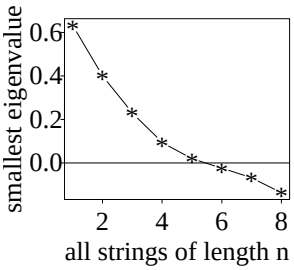
The square distance $\|\cdot\|^2$ is negative definite, thus by Theorem 11, $d_e = (\|\cdot\|^2)^{1/2}$ is also negative definite.

Assume now that $|\Sigma| > 1$. We show that $\exp(-d_e)$ is not PDS. By theorem 12, this implies that d_e is not negative definite. Let $x_1, \cdots, x_{2^n}$ be any ordering of the strings of length n over the alphabet $\{a, b\}$. Define the matrix M_n by:

$$M_n = (\exp(-d_e(x_i, x_j)))_{1 \leq i,j, \leq 2^n} \tag{30}$$

Figure 4(a) shows the smallest eigenvalue α_n of M_n as a function of n. Clearly, there are values of n for which $\alpha_n < 0$, thus the edit-distance is not negative definite. Table 4(b) provides a simple example with five strings of length 3 over the alphabet $\Sigma = \{a, b, c, d\}$ showing directly that the edit-distance is not negative definite. Indeed, it is easy to verify that: $\sum_{i=1}^{5} \sum_{j=1}^{5} c_i c_j K(x_i, x_j) = \frac{2}{3} > 0$. ■

To our knowledge, this is the first statement and proof of the fact that d_e is not NDS for $|\Sigma| > 1$. This result has a direct consequence on the design of kernels in computational biology, often based on the edit-distance or other related similarity measures. The edit-distance and other related similarity measures are often used in computational biology. When $|\Sigma| > 1$, Proposition 13 shows that d_e is not NDS. Thus, there exists $t > 0$ for which $\exp(-td_e)$ is not PDS. Similarly, d_e^2 is not NDS since otherwise by Theorem 11, $d_e = (d_e^2)^{1/2}$ would be NDS.



(a)

i	1	2	3	4	5
x_i	abc	bad	dab	adc	bcd
c_i	1	1	$-\frac{2}{3}$	$-\frac{2}{3}$	$-\frac{2}{3}$

(b)

Figure 4: (a) Smallest eigenvalue of the matrix $M_n = (\exp(-d_e(x_i, x_j)))_{1 \leq i,j, \leq 2^n}$ as a function of n. (b) Example demonstrating that the edit-distance is not negative definite.

6.2 Haussler's Convolution Kernels for Strings

D. Haussler describes a class of kernels for strings built by applying iteratively *convolution kernels* (Haussler, 1999). We show that these convolution kernels for strings are specific instances of rational kernels. Haussler (1999) defines the *convolution* of two string kernels K_1 and K_2 over the alphabet Σ as the kernel denoted by $K_1 \star K_2$ and defined for all $x, y \in \Sigma^*$ by:

$$K_1 \star K_2(x, y) = \sum_{x_1 x_2 = x, y_1 y_2 = y} K_1(x_1, y_1) \cdot K_2(x_2, y_2) \tag{31}$$

Clearly, when K_1 and K_2 are given by weighted transducers over the probability semiring, this definition coincides with that of the product (or concatenation) of transducers (Equation 3). Haussler (1999) also introduces for $0 \leq \gamma < 1$ the γ-infinite iteration of a mapping $H : \Sigma^* \times \Sigma^* \to \mathbb{R}$ by:

$$H^*_\gamma = (1 - \gamma) \sum_{n=1}^{\infty} \gamma^{n-1} H^{(n)} \tag{32}$$

where $H^{(n)} = H \star H^{(n-1)}$ is the result of the convolution of H with itself $n - 1$ times. Note that $H^*_\gamma = 0$ for $\gamma = 0$.

Lemma 14 *For $0 < \gamma < 1$, the γ-infinite iteration of a rational transduction $H : \Sigma^* \times \Sigma^* \to \mathbb{R}$ can be defined in the following way with respect to the Kleene $\dagger$-operator:*

$$H^*_\gamma = \frac{1 - \gamma}{\gamma} (\gamma H)^\dagger \tag{33}$$

Proof Haussler's convolution simply corresponds to the product (or concatenation) in the case of rational transductions. Thus, for $0 < \gamma < 1$, by definition of the $\dagger$-operator:

$$(\gamma H)^\dagger = \sum_{n=1}^{\infty} (\gamma H)^n = \sum_{n=1}^{\infty} \gamma^n H^n = \frac{\gamma}{1 - \gamma} \sum_{n=1}^{\infty} (1 - \gamma) \gamma^{n-1} H^n = \frac{\gamma}{1 - \gamma} H^*_\gamma$$

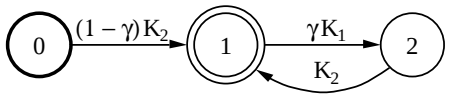


Figure 5: Haussler's convolution kernels K_H for strings: specific instances of rational kernels. K_1, (K_2), corresponds to a specific weighted transducer over the probability semiring and modeling substitutions (resp. insertions).

■

Given a probability distribution p over all symbols of Σ, Haussler's convolution kernels for strings are defined by:

$$K_H(x, y) = \gamma K_2 \star (K_1 \star K_2)^\star_\gamma + (1 - \gamma)K_2$$

where K_1 is the specific polynomial PDS rational transduction over the probability semiring defined by: $K_1(x, y) = \sum_{a\in\Sigma} p(x|a)p(y|a)p(a)$ and models substitutions, and K_2 another specific PDS rational transduction over the probability semiring modeling insertions.

Proposition 15 *For any $0 \leq \gamma < 1$, Haussler's convolution kernels K_H coincide with the following special cases of rational kernels:*

$$K_H = (1 - \gamma)[K_2(\gamma K_1 K_2)^*] \tag{34}$$

Proof As mentioned above, Haussler's convolution simply corresponds to concatenation in this context. When $\gamma = 0$, by definition, K_H is reduced to K_2 which is a rational transducer and the proposition's formula above is satisfied. Assume now that $\gamma \neq 0$. By lemma 14, K_H can be re-written as:

$$\begin{aligned} K_H &= \gamma K_2 (K_1 K_2)^\star_\gamma + (1 - \gamma)K_2 = \gamma K_2 \frac{1 - \gamma}{\gamma} (\gamma K_1 K_2)^\dagger + (1 - \gamma)K_2 \\ &= (1 - \gamma)[K_2(\gamma K_1 K_2)^\dagger + K_2] = (1 - \gamma)[K_2(\gamma K_1 K_2)^*] \end{aligned} \tag{35}$$

Since rational transductions are closed under rational operations, K_H also defines a rational transduction. Since K_1 and K_2 are PDS kernels, by theorem 6, K_H defines a PDS kernel. ■

The transducer of Figure 5 illustrates the convolution kernels for strings proposed by Haussler. They correspond to special cases of rational kernels whose mechanism is clarified by the figure: the kernel corresponds to an insertion with weight $(1 - \gamma)$ modeled by K_2 followed by any number of sequences of substitutions modeled by K_1 and insertions modeled by K_2 with weight γ. Clearly, there are many other ways of defining kernels based on weighted transducers with more complex definitions and perhaps more data-driven definitions.

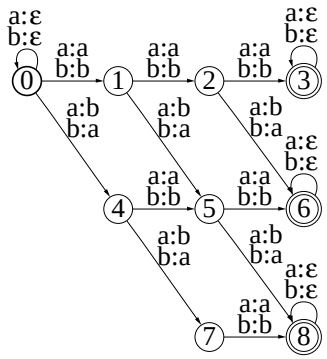


Figure 6: Mismatch kernel $K_{(k,m)} = T_{k,m} \circ T_{k,m}^{-1}$ (Leslie et al., 2003) with $k = 3$ and $m = 2$ and with $\Sigma = \{a, b\}$. The transducer $T_{3,2}$ defined over the probability semiring is shown. All transition weights and final weights are equal to one. Note that states 3, 6, and 8 of the transducer are equivalent and thus can be merged and similarly that states 2 and 5 can then be merged as well.

6.3 Other Kernels Used in Computational Biology

In this section we show the relationship between rational kernels and another class of kernels used in computational biology.

A family of kernels, *mismatch string kernels*, was introduced by (Leslie et al., 2003) for protein classification using SVMs. Let Σ be a finite alphabet, typically that of amino acids for protein sequences. For any two sequences $z_1, z_2 \in \Sigma^*$ of same length ($|z_1| = |z_2|$), we denote by $d(z_1, z_2)$ the total number of mismatching symbols between these sequences. For all $m \in \mathbb{N}$, we define the bounded distance d_m between two sequences of same length by:

$$d_m(z_1, z_2) = \begin{cases} 1 & \text{if } (d(z_1, z_2) \leq m) \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

and for all $k \in \mathbb{N}$, we denote by $F_k(x)$ the set of all factors of x of length k:

$$F_k(x) = \{z : x \in \Sigma^* z \Sigma^*, |z| = k\}$$

For any $k, m \in \mathbb{N}$ with $m \leq k$, a (k, m)-mismatch kernel $K_{(k,m)} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ is the kernel defined over protein sequences $x, y \in \Sigma^*$ by:

$$K_{(k,m)}(x, y) = \sum_{z_1 \in F_k(x),\, z_2 \in F_k(y),\, z \in \Sigma^k} d_m(z_1, z)\; d_m(z, z_2) \tag{37}$$

Proposition 16 *For any $k, m \in \mathbb{N}$ with $m \leq k$, the (k, m)-mismatch kernel $K_{(k,m)} : \Sigma^* \times \Sigma^* \to \mathbb{R}$ is a PDS rational kernel.*

Proof Let M, S, and D be the weighted transducers over the probability semiring defined by:

$$M = \sum_{a \in \Sigma} (a, a) \qquad S = \sum_{a \neq b} (a, b) \qquad D = \sum_{a \in \Sigma} (a, \epsilon) \tag{38}$$

M associates weight 1 to each pair of identical symbols of the alphabet Σ, S associates 1 to each pair of distinct or mismatching symbols, and D associates 1 to all pairs with second element ϵ.

For $i, k \in \mathbb{N}$ with $0 \leq i \leq k$, Define the *shuffle* of S^i and M^{k-i}, denoted by $S^i \sqcup\!\sqcup M^{k-i}$, as the the sum over all products made of factors S and M with exactly i factors S and $k-i$ factors M. As a finite sum of products of S and M, $S^i \sqcup\!\sqcup M^{k-i}$ is rational. Since weighted transducers are closed under rational operations the following defines a weighted transducer T over the probability semiring for any $k, m \in \mathbb{N}$ with $m \leq k$: $T_{k,m} = D^* R D^*$ with $R = \sum_{i=0}^{m} S^i \sqcup\!\sqcup M^{k-i}$. Consider two sequences z_1, z_2 such that $|z_1| = |z_2| = k$. By definition of M and S and the shuffle product, for any i, with $0 \leq i \leq m$,

$$[\![S^i \sqcup\!\sqcup M^{k-i}]\!](z_1, z_2) = \begin{cases} 1 & \text{if } (d(z_1, z_2) = i) \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

$$\text{Thus,} \quad [\![R]\!](z_1, z_2) = \sum_{i=0}^{m} S^i \sqcup\!\sqcup M^{k-i}(z_1, z_2) = \begin{cases} 1 & \text{if } (d(z_1, z_2) \leq m) \\ 0 & \text{otherwise} \end{cases}$$
$$= d_m(z_1, z_2)$$

By definition of the product of weighted transducers, for any $x \in \Sigma^*$ and $z \in \Sigma^k$,

$$T_{k,m}(x, z) = \sum_{x=uvw, z=u'v'w'} [\![D^*]\!](u, u')\ [\![R]\!](v, v')\ [\![D^*]\!](w, w') \tag{40}$$
$$= \sum_{v \in F_k(x), z=v'} [\![R]\!](v, v') = \sum_{v \in F_k(x)} d_m(v, z)$$

It is clear from the definition of $T_{k,m}$ that $T_{k,m}(x, z) = 0$ for all $x, z \in \Sigma^*$ with $|z| > k$. Thus, by definition of the composition of weighted transducer, for all $x, y \in \Sigma^*$

$$[\![T_{k,m} \circ {T_{k,m}}^{-1}]\!](x, y) = \sum_{z_1 \in F_k(x),\, z_2 \in F_k(y),\, z \in \Sigma^*} d_m(z_1, z)\ d_m(z, z_2) \tag{41}$$
$$= \sum_{z_1 \in F_k(x),\, z_2 \in F_k(y),\, z \in \Sigma^k} d_m(z_1, z)\ d_m(z, z_2) = K_{(k,m)}(x, y)$$

By proposition 7, this proves that $K_{(k,m)}$ is a PDS rational kernel. ■

Figure 6 shows $T_{3,2}$, a simple weighted transducer over the probability semiring that can be used to compute the mismatch kernel $K_{(3,2)} = T_{3,2} \circ {T_{3,2}}^{-1}$. Such transducers provide a compact representation of the kernel and are very efficient to use with the composition algorithm already described in (Cortes et al., 2003c). The transitions of these transducers can be defined implicitly and expanded on-demand as needed for the particular input strings or weighted automata. This substantially reduces the space needed for their representation, e.g., a single transition with labels $x : y$, $x \neq y$ can be used to represent all transitions with similar labels ($(a : b)$, $a, b \in \Sigma$, with $a \neq b$). Similarly, composition can also be performed on-the-fly. Furthermore, the transducer of Figure 6 can be made more compact since it admits several states that are equivalent.

7. Applications and Experiments

Rational kernels can be used in a variety of applications ranging from computational biology to optical character recognition. We have applied them successfully to a number of speech processing tasks including the identification from speech of traits, or *voice signatures*, such as emotion (Shafran et al., 2003). This section describes some of our most recent applications to spoken-dialog classification.

We first introduce a general family of PDS rational kernels relevant to spoken-dialog classification tasks that we used in our experiments, then discuss the spoken-dialog classification problem and report our experimental results.

7.1 A General Family of PDS Kernels: n-gram Kernels

A rational kernel can be viewed as a similarity measure between two sequences or weighted automata. One may for example consider two utterances to be similar when they share many common n-gram subsequences. The exact transcriptions of the utterances are not available but we can use the *word lattices* output by the recognizer instead.

A word lattice is a weighted automaton over the log semiring that compactly represents the most likely transcriptions of a speech utterance. Each path of the automaton is labeled with a sequence of words whose weight is obtained by adding the weights of the constituent transitions. The weight assigned by the lattice to a sequence of words can often be interpreted as the log-likelihood of that transcription based on the models used by the recognizer. More generally, the weights are used to rank possible transcriptions, the sequence with the lowest weight being the most favored transcription.

A word lattice A can be viewed as a probability distribution P_A over all strings $s \in \Sigma^*$. Modulo a normalization constant, the weight assigned by A to a string x is $[\![A]\!](x) = -\log P_A(x)$. Denote by $|s|_x$ the number of occurrences of a sequence x in the string s. The expected count or number of occurrences of an n-gram sequence x in s for the probability distribution P_A is:

$$c(A, x) = \sum_s P_A(s)|s|_x$$

Two lattices output by a speech recognizer can be viewed as similar when the sum of the product of the expected counts they assign to their common n-gram sequences is sufficiently high. Thus, we define an n-gram kernel k_n for two lattices A_1 and A_2 by:

$$k_n(A_1, A_2) = \sum_{|x|=n} c(A_1, x)\, c(A_2, x) \tag{42}$$

The kernel k_n is a PDS rational kernel of type $T \circ T^{-1}$ and it can be computed efficiently.

Indeed, there exists a simple weighted transducer T that can be used to computed $c(A_1, x)$ for all n-gram sequences $x \in \Sigma^*$. Figure 7 shows that transducer in the case of bigram sequences ($n = 2$) and for the alphabet $\Sigma = \{a, b\}$. The general definition of T is:

$$T = (\Sigma \times \{\epsilon\})^* \, (\sum_{x \in \Sigma} \{x\} \times \{x\})^n \, (\Sigma \times \{\epsilon\})^* \tag{43}$$

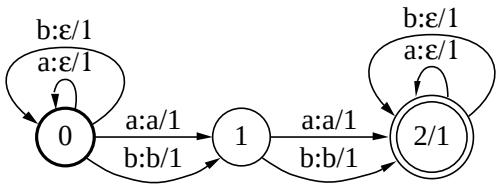


Figure 7: Weighted transducer T computing expected counts of bigram sequences of a word lattice with $\Sigma = \{a, b\}$.

k_n can be written in terms of the weighted transducer T as:

$$k_n(A_1, A_2) = w[(A_1 \circ T) \circ (T^{-1} \circ A_2)] \quad (44)$$
$$= w[(A_1 \circ (T \circ T^{-1}) \circ A_2)] \quad (45)$$

which shows that it is a rational kernel whose associated weighted transducer is $T \circ T^{-1}$. In view of Proposition 7, k_n is a PDS rational kernel. Furthermore, the general composition algorithm and shortest-distance algorithm described in Section 4 can be used to compute k_n efficiently. The size of the transducer T is $O(n|\Sigma|)$ but in practice, a lazy implementation can be used to simulate the presence of the transitions of T labeled with all elements of Σ. This reduces the size of the machine used to $O(n)$. Thus, since the complexity of composition is quadratic (Mohri et al., 1996, Pereira and Riley, 1997) and since the general shortest distance algorithm just mentioned is linear for acyclic graphs such as the lattices output by speech recognizers (Mohri, 2002), the worst case complexity of the algorithm is: $O(n^2\ |A_1|\ |A_2|)$.

By Theorem 6, the sum of two kernels k_n and k_m is also a PDS rational kernel. We define an *n-gram rational kernel* K_n as the PDS rational kernel obtained by taking the sum of all k_m, with $1 \leq m \leq n$:

$$K_n = \sum_{m=1}^{n} k_m$$

Thus, the feature space associated with K_n is the set of all m-gram sequences with $m \leq n$. n-gram kernels are used in our experiments in spoken-dialog classification.

7.2 Spoken-Dialog Classification

7.2.1 Definition

One of the key tasks of spoken-dialog systems is classification. This consists of assigning, out of a finite set, a specific category to each speech utterance based on the transcription of that utterance by a speech recognizer. The choice of possible categories depends on the dialog context considered. A category may correspond to the type of billing problem in the context of a dialog related to billing, or to the type of problem raised by the speaker in the context of a hot-line service. Categories are used to direct the dialog manager in formulating

Dataset	Number of classes	Training size	Testing size	Number of n-grams	ASR word accuracy
HMIHY 0300	64	35551	5000	24177	72.5%
VoiceTone1	97	29561	5537	22007	70.5%
VoiceTone2	82	9093	5172	8689	68.8%

Table 2: Key characteristics of the three datasets used in the experiments. The fifth column displays the total number of unigrams, bigrams, and trigrams found in the one-best output of the ASR for the utterances of the training set, that is the number of features used by BoosTexter or SVMs used with the one-best outputs.

a response to the speaker's utterance. Classification is typically based on features such as relevant key words or key sequences used by a machine learning algorithm.

The word error rate of conversational speech recognition systems is still too high in many tasks to rely only on the one-best output of the recognizer (the word error rate in the deployed services we have experimented with is about 70%, as we will see later). However, the *word lattices* output by speech recognition systems may contain the correct transcription in most cases. Thus, it is preferable to use instead the full word lattices for classification.

The application of classification algorithms to word lattices raises several issues. Even small word lattices may contain billions of paths, thus the algorithms cannot be generalized by simply applying them to each path of the lattice. Additionally, the paths are weighted and these weights must be used to guide appropriately the classification task. The use of rational kernels solves both of these problems since they define kernels between weighted automata and since they can be computed efficiently (Section 4).

7.2.2 Description of Tasks and Datasets

We did a series of experiments in several large-vocabulary spoken-dialog tasks using rational kernels with a twofold objective: to improve classification accuracy in those tasks, and to evaluate the impact on classification accuracy of the use a word lattice rather than the one-best output of the automatic speech recognition (ASR) system.

The first task we considered is that of a deployed customer-care application (HMIHY 0300). In this task, users interact with a spoken-dialog system via the telephone, speaking naturally, to ask about their bills, their calling plans, or other similar topics. Their responses to the open-ended prompts of the system are not constrained by the system, they may be any natural language sequence. The objective of the spoken-dialog classification is to assign one or several categories or call-types, e.g., *Billing Credit*, or *Calling Plans*, to the users' speech utterances. The set of categories is finite and is limited to 64 classes. The calls are classified based on the user's response to the first greeting prompt: "*Hello, this is AT&T. How may I help you?*".

Table 7.2.2 indicates the size of the HMIHY 0300 datasets we used for training and testing. The training set is relatively large with more than 35,000 utterances, this is an extension of the one we used in our previous classification experiments with HMIHY 0300 (Cortes et al., 2003c). In our experiments, we used the n-gram rational kernels described

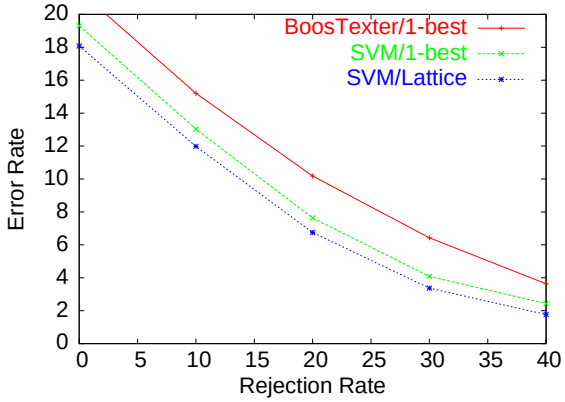


Figure 8: Classification error rate as a function of rejection rate in HMIHY 0300.

in the previous section with $n = 3$. Thus, the feature set we used was that of all n-grams with $n \leq 3$. Table 7.2.2 indicates the total number of distinct features of this type found in the datasets. The word accuracy of the system based on the best hypothesis of the speech recognizer is 72.5%. This motivated our use of the word lattices, which may contain the correct transcription in most cases. The average number of transitions of a word lattice in this task was about 260.

Table 7.2.2 reports similar information for two other datasets, VoiceTone1, and VoiceTone2. These are more recently deployed spoken-dialog systems in different areas, e.g., VoiceTone1 is a task where users interact with a system related to health-care with a larger set of categories (97). The size of the VoiceTone1 datasets we used and the word accuracy of the recognizer (70.5%) make this task otherwise similar to HMIHY 0300. The datasets provided for VoiceTone2 are significantly smaller with a higher word error rate. The word error rate is indicative of the difficulty of classification task since a higher error rate implies a more noisy input. The average number of transitions of a word lattice in VoiceTone1 was about 210 and in VoiceTone2 about 360.

Each utterance of the dataset may be labeled with several classes. The evaluation is based on the following criterion: it is considered an error if the highest scoring class given by the classifier is none of these labels.

7.2.3 Implementation and Results

We used the AT&T FSM Library (Mohri et al., 2000) and the GRM Library (Allauzen et al., 2004) for the implementation of the n-gram rational kernels K_n used. We used these kernels with SVMs, using a general learning library for large-margin classification (LLAMA), which offers an optimized multi-class recombination of binary SVMs (Haffner et al., 2003). Training time took a few hours on a single processor of a 2.4GHz Intel Pentium processor Linux cluster with 2GB of memory and 512 KB cache.

In our experiments, we used the trigram kernel K_3 with a second-degree polynomial. Preliminary experiments showed that the top performance was reached for trigram kernels and that 4-gram kernels, K_4, did not significantly improve the performance. We also found

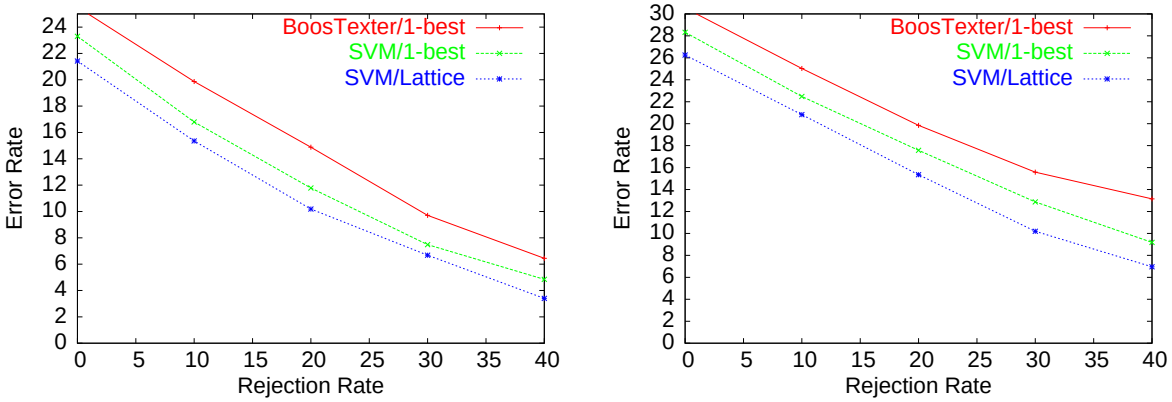


Figure 9: Classification error rate as a function of rejection rate in (a) VoiceTone1 and (b) VoiceTone2 .

that the combination of a second-degree polynomial kernel with the trigram kernel significantly improves performance over a linear classifier, but that no further improvement could be obtained with a third-degree polynomial.

We used the same kernels in the three datasets previously described and applied them to both the speech recognizer's single best hypothesis (one-best results), and to the full word lattices output by the speech recognizer. We also ran, for the sake of comparison, the BoosTexter algorithm (Schapire and Singer, 2000) on the same datasets by applying it to the one-best hypothesis. This served as a baseline for our experiments.

Figure 7.2.3 shows the result of our experiments in the HMIHY 0300 task. It gives classification error rate as a function of rejection rate (utterances for which the top score is lower than a given threshold are rejected) in HMIHY 0300 for: BoosTexter, SVM combined with our kernels when applied to the one-best hypothesis, and SVM combined with kernels applied to the full lattices.

SVM with trigram kernels applied to the one-best hypothesis leads to better classification than BoosTexter everywhere in the range of 0-40% rejection rate. The accuracy is about 2-3% absolute value better than that of BoosTexter in the range of interest for this task, which is roughly between 20% and 40% rejection rate. The results also show that the classification accuracy of SVMs combined with trigram kernels applied to word lattices is consistently better than that of SVMs applied to the one-best alone by about 1% absolute value.

Figure 7.2.3 shows the results of our experiments in the VoiceTone1 and VoiceTone2 tasks using the same techniques and comparisons. As observed previously, in many regards, VoiceTone1 is similar to the HMIHY 0300 task, and our results for VoiceTone1 are comparable to those for HMIHY 0300. The results show that the classification accuracy of SVMs combined with trigram kernels applied to word lattices is consistently better than that of BoosTexter, by more than 4% absolute value at about 20% rejection rate. They also demonstrate more clearly the benefits of the use of the word lattices for classification

in this task. This advantage is even more manifest for the VoiceTone2 task for which the speech recognition accuracy is lower. VoiceTone2 is also a harder classification task as can be seen by the comparison of the plots of Figure 7.2.3. The classification accuracy of SVMs with kernels applied to lattices is more than 6% absolute value better than that of BoosTexter near 40% rejection rate, and about 3% better than SVMs applied to the one-best hypothesis.

Thus, our experiments in spoken-dialog classification in three distinct large-vocabulary tasks demonstrates that using rational kernels with SVMs consistently leads to very competitive classifiers. They also show that their application to the full word lattices instead of the single best hypothesis output by the recognizer systematically improves classification accuracy.

8. Conclusion

We presented a general framework based on weighted transducers, rational kernels, to extend kernel methods to the analysis of variable-length sequences or more generally weighted automata. The transducer representation provides a very compact representation benefiting from existing and well-studied optimizations. It further avoids the design of special-purpose algorithms for the computation of the kernels covered by the framework of rational kernels. A single general and efficient algorithm was presented to compute effectively all rational kernels. Thus, it is sufficient to implement that algorithm and let different instances of rational kernels be given by the weighted transducers that define them. A general framework is also likely to help understand better kernels over strings or automata and their relation.

We gave the proof of several characterization results and closure properties for PDS rational kernels. These results can be used to design a complex PDS rational kernel from simpler ones or from an arbitrary weighted transducer over an appropriate semiring, or from negative definite kernels.

We also gave a study of the relation between rational kernels and several kernels or similarity measures introduced by others. Rational kernels provide a unified framework for the design of computationally efficient kernels for strings or weighted automata. The framework includes in particular pair-HMM string kernels (Durbin et al., 1998, Watkins, 1999), Haussler's convolution kernels for strings, the path kernels of Takimoto and Warmuth (2003), and other classes of string kernels introduced for computational biology. We also showed that the classical edit-distance does not define a negative definite kernel when the alphabet contains more than one symbol, a result that to our knowledge had never been stated or proved and that can guide the study of kernels for strings in computational biology and other similar applications.

Our experiments in several different large-vocabulary spoken-dialog tasks show that rational kernels can be combined with SVMs to form powerful classifiers and demonstrate the benefits of the use of kernels applied to weighted automata. There are many other rational kernels such as complex gappy n-gram kernels that could be explored and that perhaps could further improve classification accuracy in such experiments. We present elsewhere new rational kernels exploiting higher-order moments of the distribution of the counts of sequences, *moment kernels*, and report the results of our experiments on the same tasks which demonstrate a consistent gain in classification accuracy (Cortes and Mohri,

2004). Rational kernels can be used in a similar way in many other natural language processing, speech processing, and bioinformatics tasks.

Acknowledgments

We thank Allen Van Gelder, David Haussler, Risi Kondor, Alex Smola, and Manfred Warmuth for discussions about this work. We are also grateful to our colleagues of the AT&T HMIHY and VoiceTone teams for making available the data we used in our experiments.

References

Cyril Allauzen, Mehryar Mohri, and Brian Roark. A General Weighted Grammar Library. In *Proceedings of the Ninth International Conference on Automata (CIAA 2004)*, Kingston, Ontario, Canada, July 2004. URL `http://www.research.att.com/sw/tools/grm`.

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

Jean Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher: Stuttgart, 1979.

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

Bernhard E. Boser, Isabelle Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*, volume 5, pages 144–152, Pittsburg, 1992. ACM.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Lattice Kernels for Spoken Dialog Classification. In *Proceedings ICASSP'03*, Hong Kong, 2003a.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Positive Definite Rational Kernels. In *Proceedings of The 16th Annual Conference on Computational Learning Theory (COLT 2003)*, volume 2777 of *Lecture Notes in Computer Science*, pages 41–56, Washington D.C., August 2003b. Springer, Heidelberg, Germany.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational Kernels. In *Advances in Neural Information Processing Systems (NIPS 2002)*, volume 15, Vancouver, Canada, March 2003c. MIT Press.

Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Weighted Automata Kernels – General Framework and Algorithms. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech '03), Special Session Advanced Machine Learning Algorithms for Speech and Language Processing*, Geneva, Switzerland, September 2003d.

Corinna Cortes and Mehryar Mohri. Distribution Kernels Based on Moments of Counts. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML 2004)*, Banff, Alberta, Canada, July 2004.

Corinna Cortes and Vladimir N. Vapnik. Support-Vector Networks. *Machine Learning*, 20 (3):273–297, 1995.

Jack J. Dongarra, Jim R. Bunch, Cleve B. Moler, and G. W. Stewart. *LINPACK User's Guide*. SIAM Publications, 1979.

Richard Durbin, Sean Eddy, Anders Krogh, , and Graeme 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.

Patrick Haffner, Gokhan Tur, and Jeremy Wright. Optimizing SVMs for complex Call Classification. In *Proceedings ICASSP'03*, 2003.

David Haussler. Convolution Kernels on Discrete Structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, 1999.

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

Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, and Winston, 1976.

Christina Leslie, Eleazar Eskin, Jason Weston, and William Stafford Noble. Mismatch String Kernels for SVM Protein Classification. In *NIPS 2002*, Vancouver, Canada, March 2003. MIT Press.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics - Doklady*, 10:707–710, 1966.

Huma Lodhi, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS 2000*, pages 563–569. MIT Press, 2001.

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

Mehryar Mohri. Edit-Distance of Weighted Automata: General Definitions and Algorithms. *International Journal of Foundations of Computer Science*, 14(6):957–982, 2003.

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, 1996. John Wiley and Sons, Chichester.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. The Design Principles of a Weighted Finite-State Transducer Library. *Theoretical Computer Science*, 231:17–32, January 2000. URL `http://www.research.att.com/sw/tools/fsm`.

Fernando C. N. Pereira and Michael D. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*, pages 431–453. MIT Press, Cambridge, Massachusetts, 1997.

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

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

Bernhard Schölkopf. The Kernel Trick for Distances. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS 2001*, pages 301–307. MIT Press, 2001.

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

Izhak Shafran, Michael Riley, and Mehryar Mohri. Voice Signatures. In *Proceedings of The 8th IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2003)*, St. Thomas, U.S. Virgin Islands, November 2003.

Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *The Journal of Machine Learning Research (JMLR)*, 4:773 – 818, 2003.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Chris Watkins. Dynamic alignment kernels. Technical Report CSD-TR-98-11, Royal Holloway, University of London, 1999.