

Kernel Methods and Automata

Anton Xue

1 Introduction

In this sketch we study the relation between graph kernels and finite state machines. We study previous work in algebraic formulation for automata theory and kernel methods, with an interest in how these techniques may be extended to formal methods.

2 Preliminaries

2.1 Algebraic Automata Theory

Previous work in formulation in algebraic foundations for automata theory exist in literature [3], and much of our notation is taken from [2]. We overload notation for addition (+) and multiplication (\cdot) in algebraic structures when possible to avoid clutter. Similarly, the additive identity (0) and multiplicative identity (1) are also overloaded when possible.

Definition 1 (Monoid). *A monoid is an algebraic structure $(\mathbb{K}, \cdot, 1)$ where:*

- \mathbb{K} is closed under monoid multiplication $\cdot : \mathbb{K} \times \mathbb{K} \rightarrow \mathbb{K}$.
- 1 is the multiplicative identity.

When possible, we elide the \cdot in monoid multiplication to write ab instead of $a \cdot b$. When multiplication \cdot is commutative, the system is known as a commutative monoid.

Definition 2 (Semiring). *A semiring $(\mathbb{K}, +, \cdot, 0, 1)$ is a system where:*

- Semiring addition is a commutative monoid $(\mathbb{K}, +, 0)$.
- Semiring multiplication $(\mathbb{K}, \cdot, 1)$ is a monoid.
- 0 annihilates semiring multiplication.

A weighted finite state transducer (WFST) is a very general transition system defined using a semiring to specify transition weights.

Definition 3 (Weighted Finite State Transducer). *A weighted finite state transducer over a semiring \mathbb{K} is a system $(\Sigma_I, \Sigma_O, Q, I, F, \Delta, \lambda, \rho)$ where:*

- Σ_I is a finite input alphabet.
- Σ_O is a finite output alphabet.
- Q is a finite set of states.
- $I \subseteq Q$ is the set of starting states.
- $F \subseteq Q$ is the set of final states.
- $\Delta \subseteq Q \times (\Sigma_I \cup \{\varepsilon\}) \times \mathbb{K} \times (\Sigma_O \cup \{\varepsilon\}) \times Q$ is the transition function weighted by \mathbb{K} .
- $\lambda : I \rightarrow \mathbb{K}$ is the initial state weight function.
- $\rho : F \rightarrow \mathbb{K}$ is the final state weight function.

Note that the transition function Δ is defined to permit non-deterministic behavior by default.

It should be noted that Σ_I and Σ_O can be seen as the generators of the free monoids Σ_I^* and Σ_O^* , which represents the set of all strings over Σ_I and Σ_O respectively.

Although we are not yet interested in the full generality that a WFST offers, it is still nice to see what is available to us in terms of abstraction.

A special case of WFSTs that we are interested in are non-deterministic finite automata, whose specification in terms of WSFTs is by Cortes [2]. We introduce a simplified structure.

Definition 4 (Non-deterministic Finite Automata). *A non-deterministic finite automata is a system $(\Sigma, Q, \Delta, I, F)$ where:*

1. Σ is a finite alphabet.
2. Q is a finite set of states.
3. $\Delta \subseteq Q \times \Sigma \times Q$ is the transition function.
4. $I \subseteq Q$ is the set of initial states.
5. $F \subseteq Q$ is the set of final states.

We do not permit ε -transitions in our definition, which can be eliminated anyways [4].

2.2 Reproducing Kernel Hilbert Spaces

Inner products spaces are nice because they allow us to measure projection. Additionally, an inner product induces a norm, from which a metric, and thus topology, can also be created.

Definition 5 (Hilbert Space). *A Hilbert space is a inner product space that is complete with respect to the metric induced by its inner product.*

If \mathcal{H} is a Hilbert space, write $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$ to denote its inner product. We drop the subscript when context is clear, and remark that another option is for complex-valued inner products.

A special type of Hilbert spaces are known as reproducing kernel Hilbert spaces [1]. In short, these are Hilbert spaces with a special function known as the reproducing kernel.

Definition 6 (Reproducing Kernel). *Let \mathcal{H} be a Hilbert space. A reproducing kernel is a function $k : \mathcal{H} \times \mathcal{H} \rightarrow \mathbb{R}$ that satisfies the property*

$$f(x) = \langle f, k(x, \cdot) \rangle_H$$

for all $f \in \mathcal{H}$.

A reproducing Kernel hilbert space can be induced by the existence of a kernel function (not to be confused with the reproducing kernel). This type of kernel function can be defined over general sets, and provides a way of taking inner products by embedding them into a Hilbert spcae.

Definition 7 (Kernel). *Let \mathcal{X} be a set. A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ if:*

1. $k(x, y) = k(y, x)$ for all $x, y \in \mathcal{X}$. This is known as symmetric.
2. If for all $x_1, x_2, \dots, x_n \in \mathcal{X}$ the Gram matrix K defined by:

$$K_{i,j} = k(x_i, x_j)$$

is positive semi-definite.

Given a set \mathcal{X} and a kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, one can generate a Hilbert space. Define a “feautre map” $\varphi : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{X}}$ as follows:

$$\varphi(x) = k(x, \cdot)$$

Then φ maps each element of \mathcal{X} into a Hilbert space \mathcal{H} consisting of the closure of the span of functions $f : \mathcal{X} \rightarrow \mathbb{R}$:

$$\mathcal{H} = \overline{\text{span} \{f : \mathcal{X} \rightarrow \mathbb{R}\}} = \overline{\left\{ \sum_{i=1}^n a_i k(\cdot, x_i) : n \in \mathbb{N}, x_i \in \mathcal{X}, a_i \in \mathbb{R} \right\}}$$

Consider two functions $f, g \in \mathcal{H}$ which would have form:

$$f(x) = \sum_{i \in I} a_i k(x, u_i) \quad g(x) = \sum_{j \in J} b_j k(x, v_j)$$

where each $u_i, v_j \in \mathcal{X}$. Note that the summation may be infinite over the index sets I and J may be infinite, since Hilbert spaces are complete with respect to the norm induced by the inner product. The inner product between f and g is then defined as:

$$\langle f, g \rangle = \left\langle \sum_{i \in I} a_i k(\cdot, u_i), \sum_{j \in J} b_j k(\cdot, v_j) \right\rangle = \sum_{i \in I} \sum_{j \in J} a_i b_j k(u_i, v_j)$$

This is further described and proven in previous work [2, 5].

3 Automata Embeddings

4 Bounding Inner Products

References

- [1] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.
- [2] Corinna Cortes, Patrick Haffner, and Mehryar Mohri. “Rational kernels: Theory and algorithms”. In: *Journal of Machine Learning Research* 5.Aug (2004), pp. 1035–1062.
- [3] Werner Kuich and Arto Salomaa. *Semirings, automata, languages*. Vol. 5. Springer Science & Business Media, 2012.
- [4] John E Savage. *Models of computation*. Vol. 136. Addison-Wesley Reading, MA, 1998.
- [5] S Vichy N Vishwanathan et al. “Graph kernels”. In: *Journal of Machine Learning Research* 11.Apr (2010), pp. 1201–1242.