Kernels on Prolog Ground Terms

Andrea Passerini and **Paolo Frasconi**
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze
http://www.dsi.unifi.it/neural/

Abstract

We describe a family of kernels over untyped and typed Prolog ground terms and show that they can be applied for learning in structured domains, presenting experimental results in a QSPR task.

1 Introduction

Starting from the seminal work of Haussler on convolution kernels [Haussler, 1999], several researchers have proposed kernels on discrete data structures such as sequences, trees and graphs (see [Gärtner, 2003] for a review). Recently, kernels over complex individuals have been defined using higher order logic abstractions [Gärtner *et al.*, 2004]. The family of kernels developed in this paper originates from a specialization to first-order logic of such kernels, as it is designed to work on individuals represented as Prolog ground terms. There are several reasons for such a specialization. *Simplicity*: Prolog representations provide a simpler representational framework. *Sufficiency*: useful higher order structures such as sets can be simulated in Prolog. Types can also be introduced. In practice, Prolog based representations are sufficiently expressive for many application domains. *Legacy*: many inductive logic programming systems and knowledge bases are actually based on first order logic and Prolog is a well supported language. *Extendibility*: first-order logic programs have been extended to deal with uncertainty through the use of probability resulting in models such as stochastic and Bayesian logic programs (see [De Raedt and Kersting, 2003] for a review); these extensions can be very interesting in the context of machine learning.

We develop variants of the kernel on typed and untyped terms. The present formulation allows us to obtain a relatively simple proof of positive definiteness based on showing that kernels over terms are a special case of convolution kernels [Haussler, 1999]. We finally present experimental evidence of the usefulness of these kernels by learning the boiling point of alkanes from their chemical structure.

2 Kernels on Prolog ground terms

We begin with kernels on untyped terms. For a given program P we denote by $\mathcal{B}$ the Herbrand universe of P (i.e. the set of all ground terms that can be formed from symbols in P) by $\mathcal{C} \subset \mathcal{B}$ the set of all constants, and by $\mathcal{F}$ the ranked set of functors.

Definition 1 (Kernels on untyped terms). *The kernel between two terms t and s is a function $K : \mathcal{B} \times \mathcal{B} \mapsto \mathbb{R}$ defined inductively as follows:*

- *if $s \in \mathcal{C}$ and $t \in \mathcal{C}$ then $K(s,t) = \kappa(s,t)$ where $\kappa : \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}$ is a valid kernel on constants;*
- *else if s and t are compound terms and have different arities or functors, i.e. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$, then*

$$K(s,t) = \iota(f,g) \tag{1}$$

  where $\iota : \mathcal{F} \times \mathcal{F} \mapsto \mathbb{R}$ is a valid kernel on functors;
- *else if s and t are compound terms and have the same arity and functor, i.e. $s = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$, then*

$$K(s,t) = \iota(f,f) + \sum_{i=1}^{n} K(s_i, t_i) \tag{2}$$

- *in all other cases $K(s,t) = 0$.*

We call κ and ι *atomic* kernels as they operate on non-structured symbols. A special but useful case is the atomic match kernel δ defined as $\delta(x,z) = 1$ if $x = z$ and $\delta(x,z) = 0$ if $x \neq z$.

A finer level of granularity in the definition of ground term kernels can be gained from the use of typed terms. This extra flexibility may be necessary to specify different kernel functions associated with constants of different type (e.g. numerical vs. categorical). It may also be necessary in order to specify different kernels associated to different arguments of compound terms. Our approach for introducing types is similar to that proposed in [Lakshman and Reddy, 1991]. We denote by $\mathcal{T}$ the ranked set of type constructors. The type signature of a function of arity n has the form $\tau_1 \times, \ldots, \times \tau_n \mapsto \tau'$ where $n \geq 0$ is the number of arguments, $\tau_1, \ldots, \tau_k \in \mathcal{T}$ their types, and $\tau' \in \mathcal{T}$ the type of the result. Functions of arity 0 have signature $\bot \mapsto \tau'$ and can be therefore interpreted as constants of type τ'. The type signature of a predicate of arity n has the form $\tau_1 \times, \ldots, \times \tau_n \mapsto \Omega$ where $\Omega \in \mathcal{T}$ is the type of booleans. We write $t : \tau$ to assert that t is a term of type τ. We denote by $\mathcal{B}$ the set of all typed ground terms, by $\mathcal{C} \subset \mathcal{B}$ the set of all typed constants, and by $\mathcal{F}$ the set of typed functors. Finally we introduce a (possibly empty) set of *distinguished* type signatures $\mathcal{D} \subset \mathcal{T}$ that can be useful to specify ad-hoc kernel functions on certain compound terms.

Definition 2 (Kernels on typed terms). *The kernel between two typed terms t and s is defined inductively as follows:*

- *if $s \in \mathcal{C}$, $t \in \mathcal{C}$, $s : \tau$, $t : \tau$ then $K(s,t) = \kappa_\tau(s,t)$ where $\kappa_\tau : \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}$ is a valid kernel on constants of type τ;*

- *else if s and t are compound terms that have the same type but different arities, functors, or signatures, i.e. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$, $f : \sigma_1 \times, \ldots, \times \sigma_n \mapsto \tau'$, $g : \tau_1 \times, \ldots, \times \tau_m \mapsto \tau'$, then*

$$K(s,t) = \iota_{\tau'}(f,g) \tag{3}$$

where $\iota_{\tau'} : \mathcal{F} \times \mathcal{F} \mapsto I\!R$ is a valid kernel on functors that construct terms of type τ'

- *else if s and t are compound terms and have the same type, arity, and functor, i.e. $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, and $f : \tau_1 \times, \ldots, \times \tau_n \mapsto \tau'$, then*

$$K(s,t) = \begin{cases} \kappa_{\tau_1 \times, \ldots, \times \tau_n \mapsto \tau'}(s,t) \\ \quad \textit{if } (\tau_1 \times, \ldots, \times \tau_n \mapsto \tau') \in \mathcal{D} \\ \iota_{\tau'}(f,f) + \sum_{i=1}^{n} K(s_i, t_i) \quad \textit{otherwise} \end{cases} \tag{4}$$

- *in all other cases $K(s,t) = 0$.*

In the case of numerical constants, examples of useful kernels include $\kappa_{Real}(x,z) = \exp(-\gamma(x-z)^2)$ and $\kappa_{Real}(x,z) = \min\{x,z\}$.

Theorem 1. *The kernel functions on Prolog ground terms given in Definitions 1 and 2 are positive definite (pd).*

Proof sketch. Let us introduce the following decomposition structure (see [Shawe-Taylor and Cristianini, 2004]): $\mathcal{R} = \langle (X_1, X_2), R, (k_1, k_2) \rangle$ with $X_1 = \mathcal{F}$ (the set of functors), $X_2 = (\mathcal{F}, I\!N, \mathcal{B})$), and

$$R = \{(f, (f, n, a), s) \text{ s.t. } s \text{ is a term having functor } f, \text{ tuple of arguments } a, \text{ and } f \text{ has arity } n\}$$

Then it can be immediately verified that the kernel function of Eq. 1 and 2 correspond to the direct sum decomposition kernel associated with the decomposition structure $\mathcal{R}$ if $k_1 = \iota$ and $k_2((f,n,a),(g,m,b)) = \delta(f,g)\delta(n,m)k'(a,b)$ where given $a = (s_1, \ldots, s_n)$ and $b = (t_1, \ldots, t_n)$

$$k'(a,b) = \sum_{i=1}^{n} K(s_i, t_i).$$

Since kernels are closed under direct sum, if K is pd then K' is also pd. The proof then follows from Haussler's decomposition framework [Haussler, 1999] and by induction using as base step the fact that κ is a valid kernel on constants.

The proof that kernels on typed terms (Def. 2) are pd is obtained similarly, replacing functor arities by type signatures in the decomposition structure. □

Alternative forms of typed and untyped kernels can be obtained by replacing sums with products in Eq. (2) and (4), while keeping the rest of the definitions unchanged. Since kernels are closed under tensor product, positive definiteness can be proven using the same technique as in Theorem 1.

3 Experimental evaluation

The kernel has been tested on a quantitative structure-property relationship (QSPR) prediction task, namely predicting the boiling point of alkanes given their chemical structure. Alkanes can be represented as ordered rooted trees using a very simple procedure [Bianucci *et al.*, 2000], and trees can be easily encoded as Prolog ground terms, as shown in Fig. 1. We addressed this task using kernel ridge regression [Shawe-Taylor and Cristianini, 2004] with the kernel K on untyped terms of Def. 1, a match kernel for functors (carbon atoms), and a null kernel for constants (hydrogen atoms). In this way the kernel measures the number of carbon atoms in corresponding positions. As an additional source of information, we extracted the depths of the trees representing the molecules, and summed their product to the term kernel, obtaining a more informed kernel K'. Finally, we employed a Gaussian kernel on top of K and K'. Performance was evaluated by a ten fold cross validation procedure, Hyperparameters (namely, the Gaussian width and the regularization parameter), were chosen by a hold-out procedure on the training set of the first fold, and kept fixed for the successive 10 fold cross validation procedure. When using kernel K we obtained an average mean square error of 4.6 Celsius degrees while using K' the error can be reduced to 3.8 degrees. These results are comparable to those produced by the highly tuned networks developed in [Bianucci *et al.*, 2000].

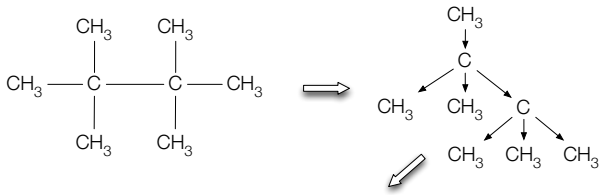

Figure 1: Ground term representation of an Alkane.

Acknowledgements

We wish to thank Luc de Raedt for fruitful discussions. This research is supported by EU Grant APrIL II (contract n° 508861) and by MIUR Grant 2003091149_002.

References

[Bianucci *et al.*, 2000] A.M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. Application of cascade correlation networks for structures to chemistry. *Appl. Intell.*, 12:117–146, 2000.

[De Raedt and Kersting, 2003] L. De Raedt and K. Kersting. Probabilistic logic learning. *SIGKDD Explor. Newsl.*, 5(1):31–48, 2003.

[Gärtner *et al.*, 2004] T. Gärtner, J.W. Lloyd, and P.A. Flach. Kernels and distances for structured data. *Machine Learning*, 57(3):205–232, 2004.

[Gärtner, 2003] T. Gärtner. A survey of kernels for structured data. *SIGKDD Explor. Newsl.*, 5(1):49–58, 2003.

[Haussler, 1999] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California, Santa Cruz, 1999.

[Lakshman and Reddy, 1991] T.K. Lakshman and U.S. Reddy. Typed prolog: A semantic reconstruction of the Mycroft-O'Keefe type system. In *Proc. of ISLP'91*.

[Shawe-Taylor and Cristianini, 2004] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.