A Reproducing Kernel Hilbert Space framework for Spike Train Signal Processing*

António R. C. Paiva, Il Park, and José C. Príncipe
Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611, USA
{arpaiva, memming, principe}@cnel.ufl.edu

February 12, 2009

Abstract

This paper presents a general framework based on reproducing kernel Hilbert spaces (RKHS) to mathematically describe and manipulate spike trains. The main idea is the definition of inner products to allow spike train signal processing from basic principles while incorporating their statistical description as point processes. Moreover, because many inner products can be formulated, a particular definition can be crafted to best fit an application. These ideas are illustrated by the definition of a number of spike train inner products. To further elicit the advantages of the RKHS framework, a family of these inner products, called the cross-intensity (CI) kernels, is further analyzed in detail. This particular inner product family encapsulates the statistical description from conditional intensity functions of spike trains. The problem of their estimation is also addressed. The simplest of the spike train kernels in this family provides an interesting perspective to other works presented in the literature, as will be illustrated in terms of spike train distance measures. Finally, as an application example, the presented RKHS framework is used to derive from simple principles a clustering algorithm for spike trains.

Keywords: Reproducing kernel Hilbert space (RKHS); spike trains; point processes; distance measures; kernel methods.

*Published in *Neural Computation*, 21(2), 424–449, February 2009, doi:10.1162/neco.2008.09-07-614.

1 Introduction

Spike trains can be observed when studying either real or artificial neurons. In neurophysiological studies, spike trains result from the activity of neurons in multiple single-unit recordings by ignoring the stereotypical shape of action potentials (Dayan and Abbott, 2001). And, more recently, there has also been a great interest in using spike trains for biologically inspired computation paradigms such as the liquid-state machine (Maass, Natschläger, and Markram, 2002; Maass and Bishop, 1998) or spiking neural networks (Bohte, Kok, and Poutré, 2002; Maass and Bishop, 1998). Regardless of the nature of the process producing the spike trains, the ultimate goal is to analyze, classify and decode the information expressed by spike trains.

A spike train $s \in \mathcal{S}(\mathcal{T})$ is a sequence of ordered spike times $s = \{t_m \in \mathcal{T} : m = 1, \ldots, N\}$ corresponding to the time instants in the interval $\mathcal{T} = [0, T]$ at which a neuron fires. Unfortunately, this formulation does not allow for the application of the usual signal processing operations to filter, eigendecompose, classify or cluster spike trains, which have been proven so useful when manipulating real world signals and form the bases to extract more information from experimental data. From a different perspective, spike trains are realizations of stochastic point processes. Therefore, they can be analyzed statistically to infer about the underlying process they represent. The main limitation in this formulation is that when multiple spike trains are analyzed they typically need to be assumed independent to avoid handling the high dimensional joint distribution.

Nevertheless, statistical analysis of spike trains is quite important, as can be asserted from the large number of methodologies that have been proposed in the literature (see Brown, Kass, and Mitra (2004) for a review). One of the fundamental descriptors of spike trains is the intensity function of the process giving rise to the observed spike train. If the spike train is assumed to be well modeled by an inhomogeneous Poisson process then many methods have been proposed for the estimation of the intensity function (Kass, Ventura, and Cai, 2003; Richmond, Optican, and Spitzer, 1990; Reiss, 1993). However, in the general case the problem is intractable since the intensity function depends on the whole history of the realization. Recently, Kass, Ventura, and Brown (2005) proposed a new spike train model simple enough to be estimated from data and yet sufficiently powerful to cope with processes more general than renewal processes. The work by Kass et al. (2005) was extended by Truccolo, Eden, Fellows, Donoghue, and Brown (2005) to allow for more general dependencies. Still, these advances depend on the availability of multiple realizations (e.g., spike trains from several trials) or spike trains of many seconds, and provide no tools to either the practitioner or the theoretician on how to analyze single realizations of multiple spike trains.

Instead, we submit that a systematic description of the theory behind single realizations

of multiple spike train analysis generalizing the methods of cross-correlation (Perkel, Gerstein, and Moore, 1967) is still needed and will enable the development of new operators for spike trains capable of transcending the results obtained with current techniques. Indeed, applying cross-correlation to spike timings is not straightforward and is the reason why, traditionally, it is applied to "binned" data. But most importantly, binning is related to instantaneous firing rate estimation and thus cross-correlation of binned spike trains cannot account for deviations from the Poisson point process model. The caveats associated with binned spike trains, in particular for temporal coding, motivated the development of methodologies involving directly spike times. This is noticeable in several spike train measures (Victor and Purpura, 1997; van Rossum, 2001; Schreiber, Fellous, Whitmer, Tiesinga, and Sejnowski, 2003) and recent attempts to use kernels to estimate and generalize these distances (Schrauwen and Campenhout, 2007). Yet, in spite of the fact that distances are very useful in classification and pattern analysis, they do not provide a suitable foundation to carry out and develop spike train signal processing algorithms.[1]

In this paper a reproducing kernel Hilbert space (RKHS) framework for spike trains is introduced with two key advantages: (1) mapping spike trains to the RKHS allows for the study of spike trains as continuous-time random functionals, thereby bypassing the limitations which lead to the use of binning, and (2) these functionals incorporate a statistical description of spike trains. In this space, a number of different signal processing algorithms to filter, eigendecompose, classify or cluster spike trains can then be developed using the linearity of the space and its inner product. Notice that, unlike approaches based on discrete representations of spike trains (such as binning) in which the dimensionality of the space becomes a problem, in the RKHS framework the dimensionality of the space is naturally dealt through the inner product.

For continuous and discrete random processes, RKHS theory has already been proven essential in a number of applications, such as statistical signal processing (Parzen, 1959) and detection (Kailath, 1971; Kailath and Duttweiler, 1972), as well as statistical learning theory (Schölkopf, Burges, and Smola, 1999; Vapnik, 1995; Wahba, 1990). Indeed, as Parzen (1959) showed, several statistical signal processing algorithms can be stated and solved easily as optimization problems in the RKHS. Although frequently overlooked, RKHS theory is perhaps an even more pivotal concept in machine learning (Schölkopf et al., 1999; Vapnik, 1995), because it is the reason for the famed kernel trick which allows for the otherwise seemingly impossible task of deriving and applying these algorithms.

In the following, we introduce a number of inner products for spike trains illustrating the generality of this methodology. We follow a systematic approach which builds the RKHS

[1] Distances define Banach spaces but for signal processing an Hilbert space (which automatically induces a Banach space) is needed.

from the ground up based on the intensity functions of spike trains, and basic requirements for the construction of an RKHS. As a result, we obtain a general and mathematically precise methodology which can yet be easily interpreted intuitively. Then, these inner products are studied in detail to show that they have the necessary properties for spike train signal processing and we propose how they can be estimated. Moreover, we discuss the RKHS and congruent spaces[2] associated with the simplest of these inner product form adding to our understanding. We then build upon this knowledge by showing that previous work in spike train measures arises naturally and effortlessly in one of the constructed RKHS, and is indeed elegantly unified in this framework. Finally, we demonstrate a practical application of the RKHS by showing how clustering of spike trains can be easily achieved using the spike train kernel.

2 Inner product for spike times

Denote the mth spike time in a spike train indexed by $i \in \mathbb{N}$ as $t_m^i \in \mathcal{T}$, with $m \in \{1, 2, \ldots, N_i\}$ and N_i the number of spike times in the spike train. To simplify the notation, however, the explicit reference to the spike train index will be omitted if is not relevant or obvious from the context.

The simplest inner product that can be defined for spike trains operates with only two spike times at a time, as observed by Carnell and Richardson (2005). In the general case, such an inner product can be defined in terms of a kernel function defined on $\mathcal{T} \times \mathcal{T}$ into the reals, with $\mathcal{T}$ the interval of spike times. Let κ denote such a kernel. Conceptually, this kernel operates in the same way as the kernels operating on data samples in machine learning (Schölkopf et al., 1999) and information theoretic learning (Príncipe, Xu, and Fisher, 2000). Although it operates only with two spike times, it will play a major role whenever we operate with complete realizations of spike trains. Indeed, the estimator for one of the spike train kernels defined next relies on this simple kernel as an elementary operation for computation or composite operations.

To take advantage of the framework for statistical signal processing provided by RKHS theory, κ is required to be a symmetric positive definite function. By the Moore-Aronszajn theorem (Aronszajn, 1950), this ensures that an RKHS $\mathcal{H}_\kappa$ exists for which κ is a reproducing kernel. The inner product in $\mathcal{H}_\kappa$ is given as

$$\kappa(t_m, t_n) = \langle \kappa(t_m, \cdot), \kappa(t_n, \cdot) \rangle_{\mathcal{H}_\kappa} = \langle \Phi_m, \Phi_n \rangle_{\mathcal{H}_\kappa} . \tag{1}$$

where Φ_m is the element in $\mathcal{H}_\kappa$ corresponding to t_m (that is, the transformed spike time).

[2]Two spaces are said to be *congruent* if there exists an isometric isomorphism, that is, a one-to-one inner product-preserving mapping, between the two spaces. This mapping is called a *congruence*.

Since the kernel operates directly on spike times and, typically, it is undesirable to emphasize events in this space, thus κ is further required to be *shift-invariant*. That is, for any $\theta \in \mathbb{R}$,

$$\kappa(t_m, t_n) = \kappa(t_m + \theta, t_n + \theta), \quad \forall t_m, t_n \in \mathcal{T}. \tag{2}$$

Hence, the kernel is only sensitive to the difference of the arguments and, consequently, we may write $\kappa(t_m, t_n) = \kappa(t_m - t_n)$.

For any symmetric, shift-invariant, and positive definite kernel, it is known that $\kappa(0) \geq |\kappa(\theta)|$.[3] This is important in establishing κ as a similarity measure between spike times. As usual, an inner product should intuitively measure some form of inter-dependence between spike times. However, the conditions posed do not restrict this study to a single kernel. On the contrary, any kernel satisfying the above requirements is theoretically valid and understood under the framework proposed here, although the practical results may vary.

An example of a family of kernels that can be used (but not limited to) are the radial basis functions (Berg, Christensen, and Ressel, 1984),

$$\kappa(t_m, t_n) = \exp(-|t_m - t_n|^p), \quad t_m, t_n \in \mathcal{T}, \tag{3}$$

for any $0 < p \leq 2$. Some well known kernels, such as the widely used Gaussian and Laplacian kernel are special cases of this family for $p = 2$ and $p = 1$, respectively.

Also of interest is to notice that for the natural norm induced by the inner product, shift-invariant kernels have the following property,

$$\|\Phi_m\| = \sqrt{\kappa(0)}, \quad \forall \Phi_m \in \mathcal{H}_\kappa. \tag{4}$$

Since the norm in $\mathcal{H}_\kappa$ of the transformed spike times point is constant, all the spike times are mapped to the surface of an hypersphere in $\mathcal{H}_\kappa$. The set of transformed spike times is called the manifold of $\mathcal{S}(\mathcal{T})$. This provides a different perspective of why the kernel used must be non-negative. Furthermore, the *geodesic distance*, corresponding to the length of the smallest path contained within the manifold (an hypersphere, in the case of shift-invariant kernels) between two points, Φ_m and Φ_n, is given by

$$\begin{aligned} d(\Phi_m, \Phi_n) &= \|\Phi_m\| \arccos\left(\frac{\langle \Phi_m, \Phi_n \rangle}{\|\Phi_m\| \, \|\Phi_n\|}\right) \\ &= \sqrt{\kappa(0)} \arccos\left[\frac{\kappa(t_m, t_n)}{\kappa(0)}\right]. \end{aligned} \tag{5}$$

Put differently, from the geometry of the transformed spike times, the kernel function is proportional to the cosine of the angle between two points in this space. Because the kernel

[3] This is a direct consequence of the fact that symmetric positive definite kernels denote inner products that obey the Cauchy-Schwarz inequality.

is non-negative, the maximum angle is $\pi/2$, which restricts the manifold of transformed spike times to a small area of the surface of the sphere. With the kernel inducing the above metric, the manifold of the transformed points forms a *Riemannian space*. This space is *not* a linear space. Its span however is obviously a linear space. In fact, it equals the RKHS associated with the kernel. Computing with the transformed points will almost surely yield points outside of the manifold of transformed spike times. This means that such points cannot be mapped back to the input space directly. This restriction however is generally not a problem since most applications deal exclusively with the projections of points in the space, and if a representation in the input space is desired it may be obtained from the projection to the manifold of transformed input points.

The kernels κ discussed this far operate with only two spike times. As in commonly done in kernel methods, kernels on spike times can be combined to define kernels that operate with spike trains. Suppose that one is interested in defining a kernel on spike trains to measure similarity in temporal spiking patterns between two spike trains (Chi and Margoliash, 2001; Chi, Wu, Haga, Hatsopoulos, and Margoliash, 2007). Such a kernel could be utilized, for example, to study temporal precision and reliability in neural spike trains in response to stimulus, or detect/classify these stimuli. This kernel could be defined as

$$V(s_i, s_j) = \begin{cases} \max\limits_{l=0,1,\ldots,(N_i-N_j)} \sum\limits_{n=1}^{N_j} \kappa(t^i_{n+l} - t^j_n), & N_i \geq N_j \\ \max\limits_{l=0,1,\ldots,(N_j-N_i)} \sum\limits_{n=1}^{N_i} \kappa(t^i_n - t^j_{n+l}), & N_i < N_j. \end{cases} \tag{6}$$

Basically, this kernel measures if spike trains have a one-to-one correspondence of the sequence of spike times. This occurs if spike trains occur with high precision and high reliability. Since spike trains are defined here in terms of fixed duration, the maximum operation in the definition searches for the best spike-to-spike correspondence. This is henceforth called the *spiking pattern matching* (SPM) kernel.

3 Inner products for spike trains

In the end of the previous section we briefly illustrated in the SPM kernel how inner products for spike trains can be built from kernels for spike times as traditionally done in machine learning. Obviously, many other spike train kernels that operate directly from data characteristics could be defined for diverse applications in a similar manner. However, in doing so it is often unclear what is the statistical structure embodied or point process model assumed by the kernel.

Rather than doing this directly, in this section, we first define general inner products for spike trains from the intensity functions, which are fundamental statistical descriptors

of the point processes. This bottom-up construction of the kernels for spike trains is unlike the previous approach taken in the previous section and is rarely taken in machine learning, but it provides direct access to the properties of the kernels defined and the RKHS they induce. In other words, in the methodology presented in this section we focus on the inner product as a *statistical descriptor*, and only then derive the corresponding estimators from data.

A spike train can be interpreted as a realization of an underlying stochastic point process (Snyder, 1975). In general, to completely characterize a point process the conditional intensity function $\lambda(t|H_t)$ is needed, where $t \in \mathcal{T} = [0, T]$ denotes the time coordinate and H_t is the history of the process up to time t. Notice that, to be a well defined function of time, $\lambda(t|H_t)$ requires a realization (so that H_t can be established), as always occurs when dealing with spike trains. This shall be implicitly assumed henceforth.

Consider two spike trains, $s_i, s_j \in \mathcal{S}(\mathcal{T})$, with $i, j \in \mathbb{N}$, and denote the corresponding conditional intensity functions of the underlying point processes by $\lambda_{s_i}(t|H_t^i)$ and $\lambda_{s_j}(t|H_t^j)$, respectively. Because of the finite duration of spike trains and the boundedness of the intensity functions, we have that

$$\int_{\mathcal{T}} \lambda^2(t|H_t)dt < \infty. \tag{7}$$

In words, conditional intensity functions are square integrable functions on $\mathcal{T}$ and, as a consequence, are valid elements of an $L_2(\mathcal{T})$ space. Obviously, the space spanned by the conditional intensity functions, denoted $L_2(\lambda_{s_i}(t|H_t^i), t \in \mathcal{T})$, is contained in $L_2(\mathcal{T})$. Therefore, we can easily define an inner product of intensity functions in $L_2(\lambda_{s_i}(t|H_t^i), t \in \mathcal{T})$ as the usual inner product in $L_2(\mathcal{T})$,

$$\begin{aligned} \mathcal{I}(s_i, s_j) &= \left\langle \lambda_{s_i}(t|H_t^i), \lambda_{s_j}(t|H_t^j) \right\rangle_{L_2(\mathcal{T})} \\ &= \int_{\mathcal{T}} \lambda_{s_i}(t|H_t^i)\lambda_{s_j}(t|H_t^j)dt. \end{aligned} \tag{8}$$

Although we defined the inner product in the space of intensity functions, it is in effect a function of two spike trains (or the underlying point processes) and thus is a kernel function in the space of spike trains. The advantage in defining the inner product from the intensity functions is that the resulting kernel incorporates the statistics of the processes directly. Moreover, the defined kernel can be utilized with *any* point process model since the conditional intensity function is a complete characterization of the point process (Cox and Isham, 1980).

The dependence of the conditional intensity functions on the history of the process renders estimation of the previous kernel intractable from finite data, as occurs in applications. A possibility is to consider a simplification of the conditional intensity functions as,

$$\lambda(t|H_t) = \lambda(t, t - t_*), \tag{9}$$

where t_* is the spike time immediately preceding t. This restricted form gives rise to inhomogeneous Markov interval (IMI) processes (Kass and Ventura, 2001). In this way it is possible to estimate the intensity functions from spike trains, and then utilize the above inner product definition to operate with them. This view is very interesting to enhance the present analysis of spike trains, but since we aim to compare the general principles presented to more typical approaches it will not be pursued in this paper.

Another way to deal with the memory dependence is to take the expectation over the history of the process H_t which yields an intensity function solely depending on time. That is,

$$\lambda_{s_i}(t) = E_{H_t^i}\left\{\lambda_{s_i}(t|H_t^i)\right\}. \tag{10}$$

This expression is a direct consequence of the general limit theorem for point processes (Snyder, 1975), and is the reason why, for example, the combined set of spike trains corresponding to multiple trials is quite well modeled as a Poisson process (Kass and Ventura, 2001). An alternate perspective is to merely assume Poisson processes to be a reasonable model for spike trains. The difference between the two perspectives is that in the second case the intensity functions can be estimated from single realizations in a plausible manner. In any case, the kernel becomes simply

$$I(s_i, s_j) = \int_{\mathcal{T}} \lambda_{s_i}(t)\lambda_{s_j}(t)dt. \tag{11}$$

Starting from the most general definition of inner product we have proposed several kernels from constrained forms of conditional intensity functions for use in applications. One can think that the definition of equation (8) gives rise to a family of cross-intensity kernels defined explicitly as an inner product, as is important for signal processing. Specific kernels are obtained from equation (8) by imposing some particular form on how to account to the dependence on the history of the process and/or allowing for a nonlinear coupling between spike trains. Two fundamental advantages of the construction methodology is that it is possible to obtain a continuous functional space where no binning is necessary and that the generality of the approach allows for inner products to be crafted to fit a particular problem that one is trying to solve.

The kernels defined so far in this section are linear operators in the space spanned by the intensity functions and are the ones that relate the most with the present analysis methods for spike trains. However, kernels between spike trains can be made nonlinear by introducing a nonlinear weighting between the intensity functions in the inner product. With this approach additional information can be extracted from the data since the nonlinearity implicitly incorporates in the measurement higher-order couplings between the intensity functions. This is of especial importance for the study of doubly-stochastic point processes,

as some theories of the brain function have put forward (Lowen and Teich, 2005). The methodology followed however is general and can be easily extended.

By analogy to how the Gaussian kernel is obtained from the Euclidean norm, we can define a similar kernel for spike trains as

$$\mathcal{I}^*_\sigma(s_i, s_j) = \exp\left[-\frac{\left\|\lambda_{s_i} - \lambda_{s_j}\right\|^2}{\sigma^2}\right], \tag{12}$$

where σ is the kernel size parameter and $\left\|\lambda_{s_i} - \lambda_{s_j}\right\| = \sqrt{\left\langle \lambda_{s_i} - \lambda_{s_j}, \lambda_{s_i} - \lambda_{s_j} \right\rangle}$ is the norm naturally induced by the inner product. This kernel is clearly nonlinear on the space of the intensity functions. On the other hand, the nonlinear mapping induced by this kernel does not operate directly on the intensity functions but on their norm and inner product, and thus has reduced descriptive ability on the coupling of their time-structure.

An alternative nonlinear CI kernel definition for spike trains is

$$\mathcal{I}^\dagger_\sigma(s_i, s_j) = \int_{\mathcal{T}} \mathcal{K}_\sigma\left(\lambda_{s_i}(t), \lambda_{s_j}(t)\right) dt, \tag{13}$$

where $\mathcal{K}_\sigma$ is a symmetric positive definite kernel with kernel size parameter σ. The advantage of this definition is that the kernel measures nonlinear couplings between the spike trains time-structure expressed in the intensity functions. In what follows, we shall refer to the definition in equation (13) as the nCI kernel. Notice that either of these nonlinear CI kernels can be made to account more detailed models of point processes.

From the suggested definitions, the *memoryless cross-intensity* (mCI) *kernel* given in equation (11) clearly adopts the simplest form since the influence of the history of the process is neglected by the kernel. This simple kernel defines an RKHS that is equivalent to cross-correlation analysis so widespread in spike train analysis (Paiva, Park, and Príncipe, 2008a), but this derivation clearly shows that it is the simplest of the cases. Still, it fits the goal of this paper as an example of the RKHS framework since it provides an interestingly broad perspective to several other works presented in the literature and suggests how methods can be reformulated to operate directly with spike trains, as will be shown next.

4 Analysis of cross-intensity kernels

4.1 Properties

In this section we present some relevant properties of the CI kernels defined in the general form of equation (8). In addition to the knowledge they provide, they are necessary for establishing that the CI kernels are well defined, induce an RKHS, and aid in the understanding of the following sections.

Property 1. *CI kernels are symmetric, non-negative and linear operators in the space of the intensity functions.*

Because the CI kernels operate on elements of $L_2(\mathcal{T})$ and corresponds to the usual dot product from L_2, this property is a direct consequence of the properties inherited. More specifically, this property guaranties the CI kernels are valid inner products.

Property 2. *For any set of $n \geq 1$ spike trains, the CI kernel matrix*

$$\boldsymbol{\mathcal{I}} = \begin{bmatrix} \mathcal{I}(s_1, s_1) & \mathcal{I}(s_1, s_2) & \dots & \mathcal{I}(s_1, s_n) \\ \mathcal{I}(s_2, s_1) & \mathcal{I}(s_2, s_2) & \dots & \mathcal{I}(s_2, s_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{I}(s_n, s_1) & \mathcal{I}(s_n, s_2) & \dots & \mathcal{I}(s_n, s_n) \end{bmatrix},$$

is symmetric and non-negative definite.

The proof is given in appendix A. Through the work of Moore (1916) and due to the Moore-Aronszajn theorem (Aronszajn, 1950), the following two properties result as corollaries of property 2.

Property 3. *CI kernels are symmetric and positive definite kernels. Thus, by definition, for any set of $n \geq 1$ point processes and corresponding n scalars $a_1, a_2, \dots, a_n \in \mathbb{R}$,*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \mathcal{I}(s_i, s_j) \geq 0. \tag{14}$$

Property 4. *There exists an Hilbert space for which a CI kernel is a reproducing kernel.*

Actually, property 3 can be obtained explicitly by verifying that the inequality of equation (14) is implied by equations (30) and (31) in the proof of property 2 (appendix A).

Properties 2 through 4 are equivalent in the sense that any of these properties implies the other two. In our case, property 2 is used to establish the other two. The most important consequence of these properties, explicitly stated through property 4, is that a *CI kernel induces a unique RKHS*, denoted in general by $\mathcal{H}_{\mathcal{I}}$. In the particular case of the mCI kernel the RKHS is denoted $\mathcal{H}_I$.

Property 5. *The CI kernels verify the Cauchy-Schwarz inequality,*

$$\mathcal{I}^2(s_i, s_j) \leq \mathcal{I}(s_i, s_i)\mathcal{I}(s_j, s_j) \qquad \forall s_i, s_j \in \mathcal{S}(\mathcal{T}). \tag{15}$$

As before, the proof is given in appendix A.

Properties 2 through 5 can also be easily proved for the nonlinear CI kernels. For the definition in equation (12), the results in Berg et al. (1984, Chapter 3) can be used

to establish that the norm is symmetric negative definite and consequently that $\mathcal{I}_\sigma^*$ is a symmetric and positive definite kernel, thus proving property 3. Properties 2, 4 and 5 follow as corollaries. Similarly, for the definition in equation (13), the proof of the properties follow the same route as for the general linear CI kernel using the linearity of the RKHS associated with the scalar kernel $\mathcal{K}$.

4.2 Estimation

From the definitions, is should be clear that for evaluation of CI kernels (linear or nonlinear) one needs first to estimate the conditional intensity function from spike trains. A possible approach is the statistical estimation framework recently proposed by Truccolo et al. (2005). Briefly, it represents a spike train point process as a discrete-time time series, and then utilizes a generalized linear model (GLM) to fit a conditional intensity function to the spike train. This is done by assuming that the logarithm of the conditional intensity function has the form

$$\log \lambda_{s_i}(\hat{t}_n|H_n^i) = \sum_{m=1}^{q} \theta_m g_m(\nu_m(\hat{t}_n)), \tag{16}$$

where $\hat{t}_n$ is the nth discrete-time instant, g_m's are general transformations of independent functions $\nu_m(\cdot)$, θ_m's are the parameter of the GLM and q is the number of parameters. Thus, GLM estimation can be used under a Poisson distribution with a log link function. The terms $g_m(\nu_m(\hat{t}_n))$ are called the predictor variables in the GLM framework and, if one considers the conditional intensity to depend only linearly on the spiking history then the g_m's can be simply delays. In general the intensity can depend nonlinearly on the history or external factor such as stimuli. Based on the estimated conditional intensity function, any of the inner products introduced in section 3 can be evaluated numerically.

Although quite general, the approach by Truccolo et al. (2005) has a main drawback: since q must be larger that the average inter-spike interval a large number of parameters need to be estimated thus requiring long spike trains (> 10 seconds). Notice that estimation of the conditional intensity function without sacrificing the temporal precision requires small bins, which means that q, and therefore the duration of the spike train used for estimation, must be increased.

In the particular case of the mCI kernel, defined in equation (11), a much simpler estimator can be derived. We now focus on this case. Since we are interested in estimating the mCI kernel from single trial spike trains, and for the reasons presented before, we will assume henceforth that spike trains are realizations of Poisson processes. Then, using kernel smoothing (Dayan and Abbott, 2001; Reiss, 1993; Richmond et al., 1990) for the estimation of the intensity function we can derive an estimator for the kernel. The advantage of this route is that a statistical interpretation is available while simultaneously approaching the

problem from a practical point of view. Moreover, in this particular case the connection between the mCI kernel and κ will now become obvious.

According to kernel smoothing intensity estimation, given a spike train s_i comprising of spike times $\{t_m^i \in \mathcal{T} : m = 1, \ldots, N_i\}$ the estimated intensity function is

$$\hat{\lambda}_{s_i}(t) = \sum_{m=1}^{N_i} h(t - t_m^i), \tag{17}$$

where h is the smoothing function. This function must be non-negative and integrate to one over the real line (just like a probability density function (pdf)). Commonly used smoothing functions are the Gaussian, Laplacian and α-function, among others.

From a filtering perspective, equation (17) can be seen as a linear convolution between the filter impulse response given by $h(t)$ and the spike train written as a sum of Dirac functionals centered at the spike times. In particular, binning is nothing but a special case of this procedure in which h is a rectangular window and the spike times are first quantized according to the width of the rectangular window (Dayan and Abbott, 2001). Moreover, it is interesting to observe that intensity estimation as shown above is directly related to the problem of pdf estimation with Parzen windows (Parzen, 1962) except for a normalization term, a connection made clear by Diggle and Marron (1988).

Consider spike trains $s_i, s_j \in \mathcal{S}(\mathcal{T})$ with estimated intensity functions $\hat{\lambda}_{s_i}(t)$ and $\hat{\lambda}_{s_j}(t)$ according to equation (17). Substituting the estimated intensity functions in the definition of the mCI kernel (equation (11)) yields the estimator,

$$\hat{I}(s_i, s_j) = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \kappa(t_m^i - t_n^j), \tag{18}$$

where κ is the kernel obtained by the autocorrelation of the intensity estimation function h with itself. A well known example for h is the Gaussian function in which case κ is also the Gaussian function (with σ scaled by $\sqrt{2}$). Another example for h is the one-sided exponential function which yields κ as the Laplacian kernel. In general, if a kernel is selected first and h is assumed to be symmetric, then κ equals the autocorrelation of h and thus h can be found by evaluating the inverse Fourier transform of the square root of the Fourier transform of κ.

The accuracy of this estimator depends only on the accuracy of the estimated intensity functions. If enough data is available such that the estimation of the intensity functions can be made exact then the mCI kernel estimation error is zero. Despite this direct dependency, the estimator effectively bypasses the estimation of the intensity functions and operates directly on the spike times of the whole realization without loss of resolution and in a computationally efficient manner since it takes advantage of the typically sparse occurrence of events.

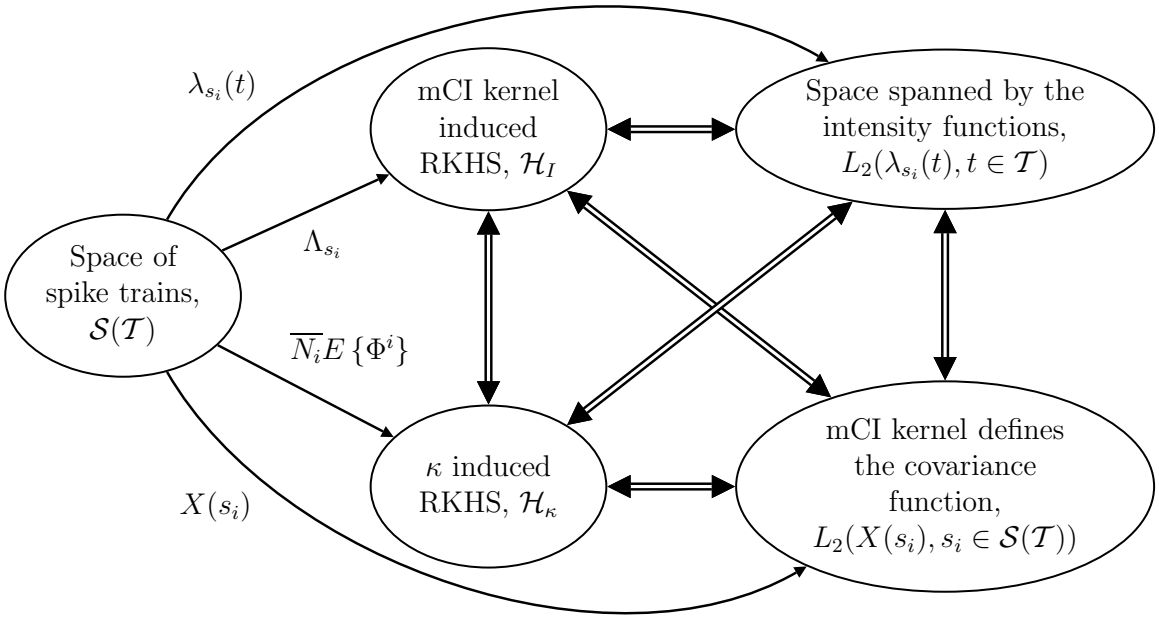


Figure 1: Relation between the original space of spike trains $\mathcal{S}(\mathcal{T})$ and the various Hilbert spaces. The double-line bi-directional connections denote congruence between spaces.

As equation (18) shows, if κ is chosen such that it satisfies the requirements in section 2, then the mCI kernel ultimately corresponds to a linear combination of κ operating on all pairwise spike time differences, one pair of spike times at a time. In other words, the mCI kernel is a linear combination of the pairwise inner products between spike times of the spike trains. Put in this way, we can now clearly see how the mCI inner product estimator builds upon the inner product on spike times presented in section 2, denoted by κ.

5 RKHS induced by the memoryless cross-intensity kernel and congruent spaces

Some considerations about the RKHS space $\mathcal{H}_I$ induced by the mCI kernel and congruent spaces are made in this section. The relationship between $\mathcal{H}_I$ and its congruent spaces provides alternative perspectives and a better understanding of the mCI kernel. Figure 1 provides a diagram of the relationships among the various spaces discussed next.

5.1 Space spanned by intensity functions

In the introduction of the mCI kernel the usual dot product in $L_2(\mathcal{T})$, the space of square integrable intensity functions defined on $\mathcal{T}$, was utilized. The definition of the inner product

in this space provides an intuitive understanding to the reasoning involved. $L_2(\lambda_{s_i}(t), t \in \mathcal{T}) \subset L_2(\mathcal{T})$ is clearly an Hilbert space with inner product defined in equation (11), and is obtained from the span of all intensity functions. Notice that this space also contains functions that are not valid intensity functions resulting from the linear span of the space (intensity functions are always non-negative). However, since our interest is mainly on the evaluation of the inner product this is of no consequence. The main limitation is that $L_2(\lambda_{s_i}(t), t \in \mathcal{T})$ is *not* an RKHS. This should be clear because elements in this space are functions defined on $\mathcal{T}$, whereas elements in the RKHS $\mathcal{H}_I$ must be functions defined on $\mathcal{S}(\mathcal{T})$.

Despite the differences, the spaces $L_2(\lambda_{s_i}(t), t \in \mathcal{T})$ and $\mathcal{H}_I$ are closely related. In fact, $L_2(\lambda_{s_i}(t), t \in \mathcal{T})$ and $\mathcal{H}_I$ are congruent. We can verify this congruence explicitly since there is clearly a one-to-one mapping,

$$\lambda_{s_i}(t) \in L_2(\lambda_{s_i}(t), t \in \mathcal{T}) \quad \longleftrightarrow \quad \Lambda_{s_i}(s) \in \mathcal{H}_I,$$

and, by definition of the mCI kernel,

$$I(s_i, s_j) = \left\langle \lambda_{s_i}, \lambda_{s_j} \right\rangle_{L_2(\mathcal{T})} = \left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle_{\mathcal{H}_I}. \tag{19}$$

A direct consequence of the basic congruence theorem is that the two spaces have the same dimension (Parzen, 1959).

5.2 Induced RKHS

In section 4.1 it was shown that the mCI kernel is symmetric and positive definite (properties 1 and 3, respectively). Consequently, by the Moore-Aronszajn theorem (Aronszajn, 1950), there exists an Hilbert space $\mathcal{H}_I$ for which the mCI kernel evaluates the inner product and is a reproducing kernel (property 4). This means that $I(s_i, \cdot) \in \mathcal{H}_I$ for any $s_i \in \mathcal{S}(\mathcal{T})$ and, for any $\zeta \in \mathcal{H}_I$, the reproducing property holds

$$\langle \zeta, I(s_i, \cdot) \rangle_{\mathcal{H}_I} = \zeta(s_i). \tag{20}$$

As a result the kernel trick follows,

$$I(s_i, s_j) = \langle I(s_i, \cdot), I(s_j, \cdot) \rangle_{\mathcal{H}_I}. \tag{21}$$

Written in this form, it is easy to verify that the point in $\mathcal{H}_I$ corresponding to a spike train $s_i \in \mathcal{S}(\mathcal{T})$ is $I(s_i, \cdot)$. In other words, given any spike train $s_i \in \mathcal{S}(\mathcal{T})$, this spike train is mapped to $\Lambda_{s_i} \in \mathcal{H}_I$, given explicitly (although unknown in closed form) as $\Lambda_{s_i} = I(s_i, \cdot)$. Then equation (21) can be restated in the more usual form as

$$I(s_i, s_j) = \left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle_{\mathcal{H}_I}. \tag{22}$$

It must be remarked that $\mathcal{H}_I$ is in fact a functional space. More specifically, that points in $\mathcal{H}_I$ are functions of spike trains; that is, they are functions defined on $\mathcal{S}(\mathcal{T})$. This is a key difference between the space of intensity functions $L_2(\mathcal{T})$ explained before and the RKHS $\mathcal{H}_I$, in that the latter allows for statistics of the transformed spike trains to be estimated as functions of spike trains.

5.3 Memoryless CI kernel and the RKHS induced by κ

The mCI kernel estimator in equation (18) shows the evaluation written in terms of elementary kernel operations on spike times. This fact alone provides an interesting perspective on how the mCI kernel uses the statistics of the spike times. To see this more clearly, consider κ to be chosen according to section 2 as a symmetric positive definite kernel, then it can be substituted by its inner product (equation (1)) in the mCI kernel estimator, yielding

$$\begin{aligned}\hat{I}(s_i, s_j) &= \sum_{m=1}^{N_i}\sum_{n=1}^{N_j}\left\langle \Phi_m^i, \Phi_n^j \right\rangle_{\mathcal{H}_\kappa} \\ &= \left\langle \sum_{m=1}^{N_i}\Phi_m^i, \sum_{n=1}^{N_j}\Phi_n^j \right\rangle_{\mathcal{H}_\kappa} .\end{aligned} \tag{23}$$

When the number of samples approaches infinity (so that the intensity functions and, consequently the mCI kernel, can be estimated exactly) the mean of the transformed spike times approaches the expectation. Hence, equation (23) results in

$$I(s_i, s_j) = \overline{N_i}\,\overline{N_j}\left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^j\right\}\right\rangle_{\mathcal{H}_\kappa}, \tag{24}$$

where $E\left\{\Phi^i\right\}$, $E\left\{\Phi^i\right\}$ denotes the expectation of the transformed spike times and $\overline{N_i}, \overline{N_j}$ are the expected number of spikes for spike trains s_i and s_j, respectively.

Equation (23) explicitly shows that the mCI kernel can be computed as a (scaled) inner product of the expectation of the transformed spike times in the RKHS $\mathcal{H}_\kappa$ induced by κ. In other words, there is a congruence $\mathscr{G}$ between $\mathcal{H}_\kappa$ and $\mathcal{H}_I$ in this case given explicitly in terms of the expectation of the transformed spike times as $\mathscr{G}(\Lambda_{s_i}) = \overline{N_i}E\left\{\Phi^i\right\}$, such that

$$\left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle_{\mathcal{H}_I} = \left\langle \mathscr{G}(\Lambda_{s_i}), \mathscr{G}(\Lambda_{s_j}) \right\rangle_{\mathcal{H}_\kappa} = \overline{N_i}\,\overline{N_j}\left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^j\right\}\right\rangle_{\mathcal{H}_\kappa}. \tag{25}$$

Recall that the transformed spike times form a manifold (the subset of an hypersphere) and, since these points have constant norm, the kernel inner product depends only on the angle between points. This is typically not true for the average of these points however. Observe that the circular variance of the transformed spike times of spike trains s_i is (Mardia and Jupp, 2000)

$$\begin{aligned}\mathrm{var}(\Phi^i) &= E\left\{\left\langle \Phi_m^i, \Phi_m^i \right\rangle_{\mathcal{H}_\kappa}\right\} - \left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^i\right\}\right\rangle_{\mathcal{H}_\kappa} \\ &= \kappa(0) - \left\|E\left\{\Phi^i\right\}\right\|^2_{\mathcal{H}_\kappa}.\end{aligned} \tag{26}$$

So, the norm of the mean transformed spike times is inversely proportional to the variance of the elements in $\mathcal{H}_\kappa$. This means that the inner product between two spike trains depends also on the dispersion of these average points. This fact is important because data reduction techniques, for example, heavily rely on optimization with the data variance. For instance, kernel principal component analysis (Schölkopf, Smola, and Müller, 1998) directly maximizes the variance expressed by equation (26) (Paiva, Xu, and Príncipe, 2006).

5.4 Memoryless CI kernel as a covariance kernel

In section 4.1 it was shown that the mCI kernel is indeed a symmetric positive definite kernel. Parzen (1959) showed that any symmetric and positive definite kernel is also a covariance function of *Gaussian distributed* random processes defined in the original space of the kernel, and the two spaces are congruent (see Wahba (1990, chapter 1) for a review). In the case of the spike train kernels defined here, this means the random processes are indexed by spike trains on $\mathcal{S}(\mathcal{T})$. This is an important result as it sets up a correspondence between the inner product due to a kernel in the RKHS to our intuitive understanding of the covariance function and associated linear statistics. Simply put, due to the congruence between the two spaces an algorithm can be derived and interpreted in any of the spaces.

Let X denote this random process. Then, for any $s_i \in \mathcal{S}(\mathcal{T})$, $X(s_i)$ is a random variable on a probability space $(\Omega, \mathcal{B}, P)$ with measure P. As proved by Parzen, this random process is Gaussian distributed with zero mean and covariance function

$$I(s_i, s_j) = E_\omega \left\{ X(s_i) X(s_j) \right\}. \tag{27}$$

Notice that the expectation is over $\omega \in \Omega$ since $X(s_i)$ is a random variable defined on Ω, a situation which can be written explicitly as $X(s_i, \omega)$, $s_i \in \mathcal{S}(\mathcal{T})$, $\omega \in \Omega$. This means that X is actually a doubly stochastic random process. An intriguing perspective is that, for any given ω, $X(s_i, \omega)$ corresponds to an ordered and almost surely non-uniform random sampling of $X(\cdot, \omega)$. The space spanned by these random variables is $L_2(X(s_i), s_i \in \mathcal{S}(\mathcal{T}))$ since X is obviously square integrable (that is, X has finite covariance).

The RKHS $\mathcal{H}_I$ induced by the mCI kernel and the space of random functions $L_2(X(s_i), s_i \in \mathcal{S}(\mathcal{T}))$ are congruent. This fact is obvious since there is clearly a congruence mapping between the two spaces. In light of this theory we can henceforward reason about the mCI kernel also as a covariance function of random variables directly dependent on the spike trains with well defined statistical properties. Allied to our familiarity and intuitive knowledge of the use of covariance (which is nothing but cross-correlation between centered random variables) this concept can be of great importance in the design of optimal learning algorithms that work with spike trains. This is because linear methods are known to be optimal for Gaussian distributed random variables.

6 Spike train distances

The concept of distance is very useful in classification and analysis of data. Spike trains are no exception. The importance of distance can be observed from the attention it has received in the literature (Victor and Purpura, 1997; van Rossum, 2001; Victor, 2005). In this section we show how the mCI kernel (or any of the presented kernels, for that matter) could be used to easily define distances between spike trains in a rigorous manner. The aim of this section is not to propose any new distance but to highlight this natural connection and convey the generality of RKHS framework by suggesting how several spike train distances can be formulated from basic principles as special cases.

6.1 Norm distance

The fact that $\mathcal{H}_I$ is an Hilbert space and therefore possesses a norm suggests an obvious definition for a distance between spike trains. In fact, since $L_2(\mathcal{T})$ is also an Hilbert space this fact would have sufficed. Nevertheless, because the inner product in $\mathcal{H}_I$ is actually evaluated in $L_2(\mathcal{T})$ the result is the same. In this sense, the distance between two spike trains or, in general, any two points in $\mathcal{H}_I$ (or $L_2(\mathcal{T})$), is defined as

$$\begin{aligned} d_{ND}(s_i, s_j) &= \left\| \Lambda_{s_i} - \Lambda_{s_j} \right\|_{\mathcal{H}_I} \\ &= \sqrt{\left\langle \Lambda_{s_i} - \Lambda_{s_j}, \Lambda_{s_i} - \Lambda_{s_j} \right\rangle_{\mathcal{H}_I}} \\ &= \sqrt{\left\langle \Lambda_{s_i}, \Lambda_{s_i} \right\rangle - 2\left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle + \left\langle \Lambda_{s_j}, \Lambda_{s_j} \right\rangle} \\ &= \sqrt{I(s_i, s_i) - 2I(s_i, s_j) + I(s_j, s_j)}. \end{aligned} \tag{28}$$

where $\Lambda_{s_i}, \Lambda_{s_i} \in \mathcal{H}_I$ denotes the transformed spike trains in the RKHS. From the properties of the norm and the Cauchy-Schwarz inequality (property 5) it immediately follows that d_{ND} is a valid distance since, for any spike trains $s_i, s_j, s_k \in \mathcal{S}(\mathcal{T})$, it satisfies the three distance axioms:

(i) Symmetry: $d_{ND}(s_i, s_j) = d_{ND}(s_j, s_i)$;

(ii) Positiveness: $d_{ND}(s_i, s_j) \geq 0$, with equality holding if and only if $s_i = s_j$;

(iii) Triangle inequality: $d_{ND}(s_i, s_j) \leq d_{ND}(s_i, s_k) + d_{ND}(s_k, s_j)$.

This distance is basically a generalization of the idea behind the Euclidean distance in a continuous space of functions.

As it was said before, this distance could be formulated directly and with the same result in $L_2(\mathcal{T})$. Then, if one considers this situation with a causal decaying exponential function as the smoothing kernel then we immediately observe that d_{ND} corresponds, in

this particular case, to the distance proposed by van Rossum (2001). Using instead a rectangular smoothing function the distance then resembles the distance proposed by Victor and Purpura (1997), as pointed by Schrauwen and Campenhout (2007), although its definition prevents a formulation in terms of the mCI kernel. Finally, using a Gaussian kernel the same distance used by Maass et al. (2002) is obtained. Notice that although it had already been noticed that other cost (i.e. kernel) functions between spike times could be used instead of the initially described (Schrauwen and Campenhout, 2007), the framework given here fully characterizes the class of valid kernels and explains their role in the time domain.

6.2 Cauchy-Schwarz distance

The previous distance is the natural definition for distance whenever an inner product is available. However, as for other L_2 spaces, alternatives measures for spike trains can be defined. In particular, based on the Cauchy-Schwarz inequality (property 5) we can define the *Cauchy-Schwarz (CS) distance* between two spike trains as

$$d_{CS}(s_i, s_j) = \arccos \frac{I(s_i, s_i)I(s_j, s_j)}{I^2(s_i, s_j)}. \tag{29}$$

From properties 1 and 5 of the mCI kernel it follows that d_{CS} is symmetric and always positive, and thus verifies the first two axioms of distance. Since d_{CS} is the angular distance between points it also verifies the triangle inequality.

The major difference between the normed distance and the CS distance is that the latter is not an Euclidean measure. Indeed, because it measures the angular distance between the spike trains it is a Riemannian metric. This utilizes the same idea expressed in equation (5) in presenting the geodesic distance associated with any symmetric positive definite kernel.

The Cauchy-Schwarz distance can be compared with the "correlation measure" between spike trains proposed by Schreiber et al. (2003). In fact, we can observe that the latter corresponds to the argument of the arc cosine and thus denotes the cosine of an angle between spike train, with norm and inner product estimated using the Gaussian kernel. Notice that Schreiber's et al. "correlation measure" is only a pre-metric since it does not verify the triangle inequality. In d_{CS} this is ensured by the arc cosine function.

7 Application example: Clustering of spike trains

Having an RKHS framework for spike trains is important because it facilitates the development of new methods to operate with spike trains. Moreover, all of these methods are developed under the same principles provided by this general theory.

To exemplify the use of kernels proposed under the RKHS framework, in the following we show how a clustering algorithm for spike trains can be obtained naturally from any of the spike train kernel definitions here presented (with results shown for the mCI and nCI kernels). Comparing these ideas with previous clustering algorithms for spike trains we find that they result in simpler methods, derived in an integrated manner, with a clear understanding of the features being accounted for, and greater generality. We remark, however, that our emphasis here is not to propose another algorithm but merely to illustrate the elegance and usefulness of the RKHS framework. For this reason, we will dispense a comparison with another methods and a more thorough analysis of the performance of the algorithm at this time.

7.1 Algorithm

For the purpose of this example, we will exemplify how spike train kernels defined in the RKHS framework provide the means to do clustering of spike trains. The algorithm is based on the ideas of spectral clustering. Spectral clustering is advantageous for the purpose of this example since the evaluation of the relation between spike trains and the actual clustering procedure is conceptually distinct. It should be possible to extend other established clustering algorithms although one must introduce the inner product directly into the computation which slightly complicates matters.

Spectral clustering of spike trains operates in two major steps. First, the affinity matrix of the spike trains is computed. Let $\{s_1, s_2, \ldots, s_n\}$ be the set of n spike trains to be clustered into k clusters. The affinity matrix is an $n \times n$ matrix describing the *similarity* between all pairs of spike trains. The second step of the algorithm is to use spectral clustering applied to this affinity matrix to find the actual clustering results. In particular, the spectral clustering algorithm proposed by Ng, Jordan, and Weiss (2001) was used for its simplicity and minimal use of parameters. We refer the reader to Ng et al. (2001) for additional details on the spectral clustering algorithm.

Clearly, the defining step for the use of this algorithm for clustering of spike trains is how to evaluate affinity between spike trains. Since inner products inherently quantify similarity, any of the kernels proposed could be used, and in particular the mCI and nCI kernels, for which we provide results. In this situation the affinity matrix is simply the Gram matrix of the spike trains computed with the spike train kernel. The algorithm shown here is similar to the method by Paiva, Rao, Park, and Príncipe (2007), but is simpler since the transformation to map the distance evaluation to a similarity measurement and the need to adjust the corresponding parameter are avoided. Since distances are derived concepts and, many times, can be defined in terms of inner products, the approach taken is much more straightforward and principled. Moreover, the computation is reduced since the spike train

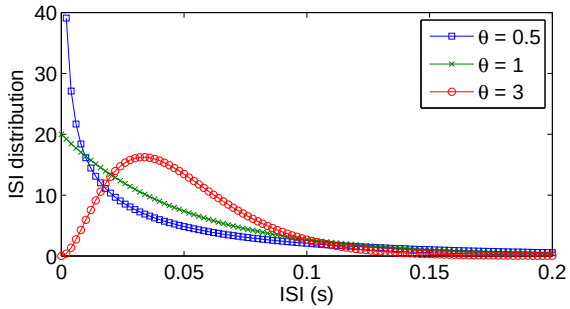
kernels are simpler to evaluate than a distance. But most importantly, the algorithm can be generalized merely by using yet another spike train kernel.

7.2 Simulation

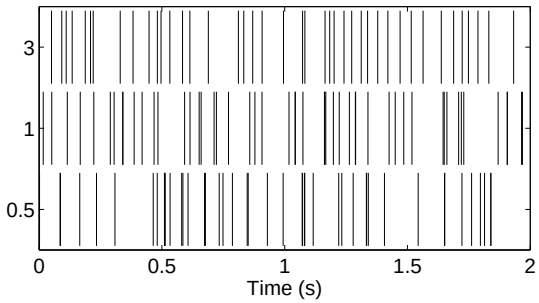
The goal of this simulation example is to show the importance of spike train kernels that go beyond the first cross-moment (i.e., cross-correlation) between spike trains. For this reason, we applied the algorithm described in the previous section for clustering of spike trains generated as homogeneous renewal point processes with a gamma inter-spike interval (ISI) distribution. This model was chosen since the Poisson process is a particular case and thus can be directly compared.

A three cluster problem is considered, in which each cluster is defined by the ISI distribution of its spike trains (see figure 2(a)). In other words, spike trains within the cluster were generated according to the same point process model. All spike trains were 1s long and with constant firing rate 20 spk/s. For each Monte Carlo run, a total of 100 spike trains randomly assigned to one of the clusters were generated. The results statistics were estimated over 500 Monte Carlo runs. For both the mCI and nCI kernels, the Gaussian function was used as smoothing function with results for three values of the smoothing width, 2, 10 and 100ms. In addition, the Gaussian kernel was utilized for $\mathcal{K}_\sigma$ in the computation of the nCI kernel, with results for kernel sizes $\sigma = 1$ and $\sigma = 10$.

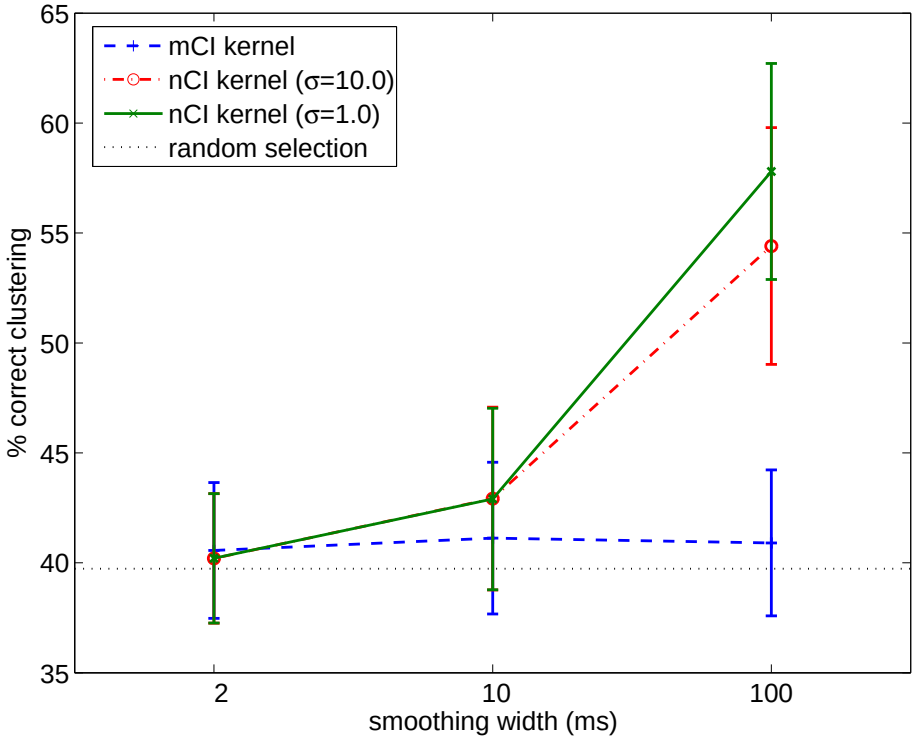
The results of the simulation are shown in figure 2(c). The cluster with shape parameter $\theta = 1$ contained Poisson spike trains, and spike trains with shape parameter $\theta = 3$ were more regular whereas $\theta = 0.5$ gave rise to more irregular (i.e. "bursty") spike trains. The results with the mCI kernel are at most 1.4% better, on average, than random selection. This low performance is not entirely surprising since all spike trains have constant firing rate. Using the nCI kernel with the larger smoothing width yielded an improvement of 14.7% for $\sigma = 10$ and 18% for $\sigma = 1$, on average. Smaller values of σ did not improve the clustering performance ($\sigma = 0.1$ resulted in the same performance as $\sigma = 1$), demonstrating that the selection of kernel size σ for the nCI kernel is not very problematic. But, most importantly, the results show that even though the formulation depends only on the memoryless intensity functions, in practice, the nonlinear kernel $\mathcal{K}_\sigma$ allows for different spike train models to be discriminated. This improvement is due to the fact that $\mathcal{K}_\sigma$ enhances the slight differences in the estimated intensity functions due to the different point process model expressed in the spike trains (cf. figure 2(b)).



(a) Inter-spike interval (ISI) distributions defining each cluster.



(b) Example spike trains from each cluster.



(c) Clustering results.

Figure 2: Comparison of clustering performance using mCI and nCI kernels for a three cluster problem.

8 Conclusion

The point process nature of spike trains has made the application of conventional signal processing methods to spike trains difficult and imprecise to apply (e.g., through binning) from first principles. The most powerful methodologies to spike train analysis are based on statistical approaches, but they face serious shortcomings with the widespread use of multielectrode array techniques since they are only practical using an independent assumption.

This paper presents a reproducing kernel Hilbert space formulation for the analysis of spike trains that has the potential to improve the set of algorithms that can be developed for spike train analysis of multielectrode array data. The paper presents the theory with sufficient detail to establish a solid foundation and hopefully entice further work along this line of reasoning. Indeed, the paper dual role is to elucidate the set of possibilities that are open by the RKHS formulation and to link a very special case of the theory to methods that are in common use in computational neuroscience. So a lot of more work is needed to bring the possibilities open by RKHS theory to fruition in spike train signal analysis.

At the theoretical level we extend the early work of Parzen (1959) on stochastic processes to spike trains by defining bottom up the structure of the RKHS on the statistics of the point process, i.e. its intensity function. This result provides a solid foundation for future work both for practical algorithm development but also on a simple way to bring into the analysis more realistic assumptions about the statistics of spike trains. Indeed we show that the Poisson statistical model is behind the simplest definition of the RKHS (the memoryless cross-intensity kernel) and that this RKHS provides a linear space for doing signal processing in spike trains. However, the same framework can be applied to inhomogeneous Markov interval of even more general point process models which only now are beginning to be explored. We would like to emphasize that building a RKHS bottom up is a much more principled approach than the conventional way that RKHS are derived in machine learning, where the link to data statistics is only possible at the level of the estimated quantities, not the statistical operators themselves.

The second theoretical contribution is to show the flexibility of RKHS methodology. Indeed it is possible to define alternate, and yet unexplored, RKHS for spike train analysis that are not linearly related to the intensity functions. Obviously that this will provide many possible avenues for future research and there is the hope that it will be possible to derive systematic approaches to tailor the RKHS definition to the goal of the data analysis. We basically see two different types of RKHS that mimic exactly the two methodologies being developed in the machine learning and signal processing literatures: kernels that are data independent (κ) and kernels that are data dependent (CI kernels). Specifically for point processes we show in a specific case how that the former may be used to compose the latter, but they work with the data in very different ways. But what is interesting is that

these two types of RKHS provide different features in the transformation to the space of functions. The former is a macroscopic descriptor of the spike time intervals that may be usable in coarse analysis of the data. The latter is a functional descriptor of the data but it is harder to compute. In computational neuroscience only the latter is being pursued, but by analogy with the large impact of kernel methods in statistical learning, we foresee an equal important impact of the former in computational neuroscience. And yet, the theory and the operators we have presented in this paper will form the foundations for such future developments.

There are also practical implications of the RKHS methodology presented in this paper. Since the RKHS is a special vector space, all the conventional signal processing algorithms that involve inner product computations can be immediately implemented in the RKHS. We illustrate this with a spectral clustering application, but many other applications are possible, ranging from filtering to eigendecompositions of spike trains in the RKHS (Paiva, Park, and Príncipe, 2008b). The spectral clustering algorithm shown could also be derived using common distances measures that have been defined for spike trains as has been done before (Paiva et al., 2007). But we stress the elegance of the proposed formulation that first defines the structure of the space (the inner product) and then leaves for the users the design of their intended algorithm, unlike the approaches presented so far which are specific for the application. It is also important to stress the computational savings for spike timing analysis provided by our RKHS methodology, which has a complexity independent to data sampling rates but only depends on the spike rates.

There are still many other topics that need to be fully researched for a systematic use of the technique. Perhaps the most important one for practical applications is the kernel size parameter of the kernel function. The theory shows clearly the role of this free parameter, i.e. it sets the scale of the transformation by changing the inner product. So it provides flexibility to the researcher, but also suggests the need to find tools to help set this parameter according to the data and the analysis goal.

Acknowledgments

The authors would like to thank Emory N. Brown for clarifying the estimation of the conditional intensity function using GLM. A. R. C. Paiva was supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004. This work was partially supported by NSF grants ECS-0422718 and CISE-0541241.

A Proofs

In this section the proofs for properties 2 and 5 given in section 4.1 are presented.

Proof of property 2. The symmetry of the matrix results immediately from property 1. By definition, a matrix is non-negative definite if and only if $\mathbf{a}^T \boldsymbol{\mathcal{I}} \mathbf{a} \geq 0$, for any $\mathbf{a}^T = [a_1, \ldots, a_n]$ with $a_i \in \mathbb{R}$. So, we have that

$$\mathbf{a}^T \boldsymbol{\mathcal{I}} \mathbf{a} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \mathcal{I}(s_i, s_j), \tag{30}$$

which, making use of the general definition for CI kernels (equation (8)), yields,

$$\begin{aligned} \mathbf{a}^T \boldsymbol{\mathcal{I}} \mathbf{a} &= \int_{\mathcal{T}} \left(\sum_{i=1}^{n} a_i \lambda_{s_i}(t|H_t^i) \right) \left(\sum_{j=1}^{n} a_j \lambda_{s_j}(t|H_t^j) \right) dt \\ &= \left\langle \sum_{i=1}^{n} a_i \lambda_{s_i}(\cdot|H_t^i), \sum_{j=1}^{n} a_j \lambda_{s_j}(\cdot|H_t^j) \right\rangle_{L_2(\mathcal{T})} \\ &= \left\| \sum_{i=1}^{n} a_i \lambda_{s_i}(\cdot|H_t^i) \right\|^2_{L_2(\mathcal{T})} \geq 0. \end{aligned} \tag{31}$$

□

Proof of property 5. Consider the 2×2 CI kernel matrix,

$$\boldsymbol{\mathcal{I}} = \begin{bmatrix} \mathcal{I}(s_i, s_i) & \mathcal{I}(s_i, s_j) \\ \mathcal{I}(s_j, s_i) & \mathcal{I}(s_j, s_j) \end{bmatrix}.$$

From property 2, this matrix is symmetric and non-negative definite. Hence, its determinant is non-negative (Harville, 1997, pg. 245). Mathematically,

$$\det(\mathbf{I}) = \mathcal{I}(s_i, s_i)\mathcal{I}(s_j, s_j) - \mathcal{I}^2(s_i, s_j) \geq 0,$$

which proves the result of equation (15). □

References

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68(3):337–404, May 1950.

C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions.* Springer-Verlag, New York, NY, 1984.

S. M. Bohte, J. N. Kok, and H. L. Poutré. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomp.*, 48(1–4):17–37, Oct. 2002. doi: 10.1016/S0925-2312(01)00658-0.

E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neurosci.*, 7:456–461, 2004. doi: 10.1038/nn1228.

A. Carnell and D. Richardson. Linear algebra for time series of spikes. In *Proc. European Symp. on Artificial Neural Networks*, pages 363–368, Bruges, Belgium, Apr. 2005.

Z. Chi and D. Margoliash. Temporal precision and temporal drift in brain and behavior of zebra finch song. *Neuron*, 32(1–20):899–910, Dec. 2001.

Z. Chi, W. Wu, Z. Haga, N. G. Hatsopoulos, and D. Margoliash. Template-based spike pattern identification with linear convolution and dynamic time warping. *J. Neurophysiol.*, 97(2):1221–1235, Feb. 2007. doi: 10.1152/jn.00448.2006.

D. R. Cox and V. Isham. *Point Processes*. Chapman and Hall, 1980.

P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, USA, 2001.

P. Diggle and J. S. Marron. Equivalence of smoothing parameter selectors in density and intensity estimation. *J. Am. Stat. Assoc.*, 83(403):793–800, Sept. 1988.

D. A. Harville. *Matrix algebra from a statistician's perspective*. Springer, 1997.

T. Kailath. RKHS approach to detection and estimation problems–part I: Deterministic signals in gaussian noise. *IEEE Trans. Inform. Theory*, 17(5):530–549, Sept. 1971.

T. Kailath and D. L. Duttweiler. An RKHS approach to detection and estimation problems–part III: Generalized innovations representations and a likelihood-ratio formula. *IEEE Trans. Inform. Theory*, 18(6):730–745, Nov. 1972.

R. E. Kass and V. Ventura. A spike-train probability model. *Neural Comp.*, 13(8):1713–1720, Aug. 2001.

R. E. Kass, V. Ventura, and C. Cai. Statistical smoothing of neuronal data. *Network: Comp. Neural Sys.*, 14:5–15, 2003.

R. E. Kass, V. Ventura, and E. N. Brown. Statistical issues in the analysis of neuronal data. *J. Neurophysiol.*, 94:8–25, 2005. doi: 10.1152/jn.00648.2004.

S. B. Lowen and M. C. Teich. *Fractal-Based Point Processes*. Wiley, 2005. ISBN 0-471-38376-7.

W. Maass and C. M. Bishop, editors. *Pulsed Neural Networks*. MIT Press, 1998.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comp.*, 14(11): 2531–2560, 2002. doi: 10.1162/089976602760407955.

K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley & Sons, West Sussex, England, 2000. ISBN 0-471-95333-4.

E. H. Moore. On properly positive Hermitian matrices. *Bull. Am. Math. Soc.*, 23:59, 1916.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances Neural Information Processing Systems*, volume 14, 2001.

A. R. C. Paiva, J.-W. Xu, and J. C. Príncipe. Kernel principal components are maximum entropy projections. In *Proc. Int. Conf. on Independent Component Analysis and Blind Source Separation, ICA-2006*, pages 846–853, Charleston, SC, Mar. 2006. doi: 10.1007/11679363_105.

A. R. C. Paiva, S. Rao, I. Park, and J. C. Príncipe. Spectral clustering of synchronous spike trains. In *Proc. IEEE Int. Joint Conf. on Neural Networks, IJCNN-2007*, Orlando, FL, USA, Aug. 2007.

A. R. C. Paiva, I. Park, and J. C. Príncipe. Reproducing kernel hilbert spaces for spike train analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-2008*, Las Vegas, NV, USA, Apr. 2008a.

A. R. C. Paiva, I. Park, and J. C. Príncipe. Optimization in reproducing kernel Hilbert spaces of spike trains. In *Computational Neuroscience*. Springer, 2008b. in press.

E. Parzen. On the estimation of a probability density function and the mode. *Annals Math. Stat.*, 33(2):1065–1076, Sept. 1962.

E. Parzen. Statistical inference on time series by Hilbert space methods. Technical Report 23, Applied Mathematics and Statistics Laboratory, Stanford University, Stanford, California, Jan. 1959.

D. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. II. simultaneous spike trains. *Biophys. J.*, 7(4):419–440, July 1967.

J. C. Príncipe, D. Xu, and J. W. Fisher. Information theoretic learning. In S. Haykin, editor, *Unsupervised Adaptive Filtering*, volume 2, pages 265–319. John Wiley & Sons, 2000.

R.-D. Reiss. *A Course on Point Processes*. Springer-Verlag, New York, NY, 1993.

B. J. Richmond, L. M. Optican, and H. Spitzer. Temporal encoding of two-dimensional patterns by single units in primate primary visual cortex. I. Stimulus-response relations. *J. Neurophysiol.*, 64(2):351–369, Aug. 1990.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comp.*, 10(5):1299–1319, 1998.

B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

B. Schrauwen and J. V. Campenhout. Linking non-binned spike train kernels to several existing spike train distances. *Neurocomp.*, 70(7–8):1247–1253, Mar. 2007. doi: 10.1016/j.neucom.2006.11.017.

S. Schreiber, J. M. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski. A new correlation-based measure of spike timing reliability. *Neurocomp.*, 52–54:925–931, June 2003. doi: 10.1016/S0925-2312(02)00838-X.

D. L. Snyder. *Random Point Process in Time and Space*. John Viley & Sons, New York, 1975.

W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiol.*, 93:1074–1089, Feb. 2005. doi: 10.1152/jn.00697.2004.

M. C. W. van Rossum. A novel spike distance. *Neural Comp.*, 13(4):751–764, 2001.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

J. D. Victor. Spike train metrics. *Current Opinion in Neurobiology*, 15(5):585–592, Sept. 2005. doi: 10.1016/j.conb.2005.08.002.

J. D. Victor and K. P. Purpura. Metric-space analysis of spike trains: theory, algorithms, and application. *Network: Comp. Neural Sys.*, 8:127–164, Oct. 1997.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1990.