Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2007) 1247–1253
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Linking non-binned spike train kernels to several existing spike train metrics

Benjamin Schrauwen*, Jan Van Campenhout

Department of Electronics and Information Systems, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium

Available online 20 December 2006

Abstract

This work presents three kernel functions that can be used as inner product operators on non-binned spike trains, allowing the use of state-of-the-art classification techniques. One of the main advantages is that this approach does not require the spike trains to be binned. Thus a high temporal resolution is preserved which is needed when temporal coding is used. The kernels are closely related to several recent and often-used spike train metrics which take into account the biological variability of spike trains. It follows that the different existing metrics are unified by the spike train kernels presented.

As a test of the classification potential of the new kernel functions, a jittered spike train template classification problem is solved.

Keywords: Spike train classification; Kernel methods; Spike train metrics; Non-binned

1. Introduction

Spike train classification, the classification of stereotypical neuron pulse trains, is of interest to neurobiological research. Here spike trains generated by real neurons need to be related to stimuli, for example in cognitive research where the function of spikes as a code is researched [14], or interpreted as in brain–machine interfaces [17]. Spike train classification is also of interest in neural network research, where spiking neurons get increasing attention [11] due to their greater computational power [10] and their inherent ability to process temporal information.

A biological spike train is modeled as a set of events containing the precise time instants at which a neuron fires. Traditionally, spike trains are represented in a simplified way by binning them: the actual spike times are grouped in so-called temporal *bins* which have a fixed size that is usually much larger than the average inter-spike-interval. This binning is founded on the rate coding hypothesis [14,1]: the actual spike times do not matter, it is the average firing rate that encodes the information. Recently however, much physiological and theoretical [19,20,23] evidence has been obtained to the effect that exact spike times *do* matter and can improve neuron processing performance greatly. The current view on this topic is that both codings are used and can co-exist [22,13]. Some parts of the brain mainly communicate through the average firing rate of their neurons (like the motor neurons), while other parts are very sensitive to the exact temporal position of the spikes (higher brain regions). Many current classification techniques and metrics for spike trains are nevertheless still based on binned spike trains.

An overview of different metrics for spike train is presented in Fig. 1. Traditionally, classification of spike trains is performed using distance metrics based on the Euclidean distance of the binned spike train [21,7] where the space dimensions equals the number of bins. However, these metrics perform poorly because they do not take into account the underlying biological variance of spike trains, such as the natural spike jitter. Recent metrics that apply to non-binned spike trains [5,12,27,15] do take these biological factors into account, which results in improved classification performance.

In this paper, we show that several of these recent metrics are identical or closely related. Furthermore we show that kernel functions can be constructed that are

*Corresponding author.
E-mail address: Benjamin.Schrauwen@UGent.be (B. Schrauwen).
URL: http://www.elis.ugent.be/SNN.

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.11.017

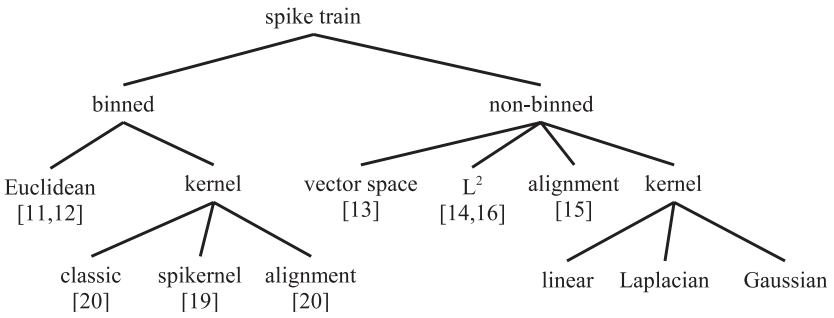

Fig. 1. Taxonomy of the different spike train metrics and their references.

closely related to these metrics. Kernel functions arise naturally in classification approaches [24,16] where measurement data $x \in X$ are mapped into a high-dimensional feature space Y using the mapping $\phi : X \to Y$. The actual classification is performed in feature space. This requires an inner product operator $\langle \phi(x), \phi(z) \rangle$ in feature space. The feature map and the inner product define a so-called kernel function $K : X \times X \to \mathbb{R}$ which equals $K(x, z) = \langle \phi(x), \phi(z) \rangle$. Usually, easy-to-compute kernel functions are directly defined in measurement space, and Mercer's theorem [24,16] guarantees the existence of a suitable implicitly defined feature mapping that does not need to be made explicit. In this paper, however, we shall explicitly identify ϕ which proves that the proposed functions indeed posses the kernel properties, and which allows us to link the kernels to several existing spike train metrics.

Kernel methods have previously already been applied to spike trains. There is the *spikernel* [18], a string-based kernel that is applied to the instantaneous firing rate, and the *alignment score kernel* [6], a kernel built by an explicit feature space mapping based on an alignment score such as the one presented in Section 4.3. However, both these methods apply to *binned* spike trains. This binning introduces an extra parameter to tune (bin size) and most importantly, throws away much of the temporal information present in the spike trains. When the information is primarily encoded in the exact spike times, these methods will not perform well.

To evaluate the performance of the spike train kernels, we use an artificial benchmark application, the jittered spike train template classification problem: two random spike patterns are generated and from these, randomly perturbed versions are created by moving spikes around. A classifier has to be trained that has to detect the spike template from which a jittered spike train originates. Although this application may appear simple, complex tasks, like speech recognition [25], are extensions of this simple benchmark.

This contribution is organized as follows. First, in Section 2, we will introduce three simple kernel functions that apply to sets of spike times. Then, in Section 3 we will show that these functions are actual kernels by presenting the actual feature mappings. In Section 4, we will present several existing non-binned spike train metrics and show that they are equivalent or strongly related to each other and strongly related to the three introduced new kernels. The classification performance of the kernels is evaluated in Section 5 on an artificial benchmark. We conclude in Section 6.

2. Spike train kernels

We will now present three kernel functions that apply to non-binned spike trains. A spike train $\mathbf{x}$ is simply the set of spike times $\mathbf{x} = \{t_1, t_2, \ldots, t_N\}$. Our kernel functions take two spike trains $\mathbf{x}$ and $\mathbf{z}$ as arguments, with N and M spike times, respectively. We denote the ith spike time of $\mathbf{x}$ as t_i and the jth spike time of $\mathbf{z}$ as t'_j.

The first kernel has a piecewise linear weighting of the spike time difference:

$$K_{\text{linear}}(\mathbf{x}, \mathbf{z}) = K_{\text{linear}}(\{t_i\}, \{t'_j\}) = \sum_{i}^{N} \sum_{j}^{M} \max\left(1 - \frac{\lambda}{2} |t_i - t'_j|, 0\right),$$

the second kernel is related to the well known Laplacian kernel:

$$K_{\text{laplace}}(\mathbf{x}, \mathbf{z}) = K_{\text{laplace}}(\{t_i\}, \{t'_j\}) = \sum_{i}^{N} \sum_{j}^{M} \exp(-\lambda |t_i - t'_j|),$$

and the third is related to the Gaussian kernel:

$$K_{\text{gauss}}(\mathbf{x}, \mathbf{z}) = K_{\text{gauss}}(\{t_i\}, \{t'_j\}) = \sum_{i}^{N} \sum_{j}^{M} \exp(-\lambda^2 (t_i - t'_j)^2).$$

All spike times of the first spike train are compared to all spike times of the second spike train and a 'weight', decreasing monotonically with the distance between the two spikes, is taken into account. This results in kernels that operate on *sets* instead of on vectors, as is usually the case. A plot of the weight functions applied to individual spike pairs is shown in Fig. 2. The scale parameter λ determines the 'width' of the kernels. A large λ leads to a kernel relating only near-by spikes, small λ results in a kernel comparing all spikes to all other spikes. The kernels are scaled such that the widths of all three kernels are approximately equal for a given λ. This will allow a more easy comparison of the kernel's performance.