Kernel Measures of Independence for non-*iid* Data*

Xinhua Zhang
NICTA and Australian National University
Canberra, Australia
xinhua.zhang@anu.edu.au

Le Song†
School of Computer Science
Carnegie Mellon University, Pittsburgh, USA
lesong@cs.cmu.edu

Arthur Gretton
MPI Tübingen for Biological Cybernetics
Tübingen, Germany
arthur@tuebingen.mpg.de

Alex Smola†
Yahoo! Research
Santa Clara, CA, United States
alex@smola.org

Abstract

Many machine learning algorithms can be formulated in the framework of statistical independence such as the Hilbert Schmidt Independence Criterion. In this paper, we extend this criterion to deal with structured and interdependent observations. This is achieved by modeling the structures using undirected graphical models and comparing the Hilbert space embeddings of distributions. We apply this new criterion to independent component analysis and sequence clustering.

1 Introduction

Statistical dependence measures have been proposed as a unifying framework to address many machine learning problems. For instance, clustering can be viewed as a problem where one strives to maximize the dependence between the observations and a discrete set of labels [15]. Conversely, if labels are given, feature selection can be achieved by finding a subset of features in the observations which maximize the dependence between labels and features [16]. Similarly in supervised dimensionality reduction [14], one looks for a low dimensional embedding which retains additional side information such as class labels. Likewise, blind source separation (BSS) tries to unmix independent sources, which requires a contrast function quantifying the dependence of the unmixed signals.

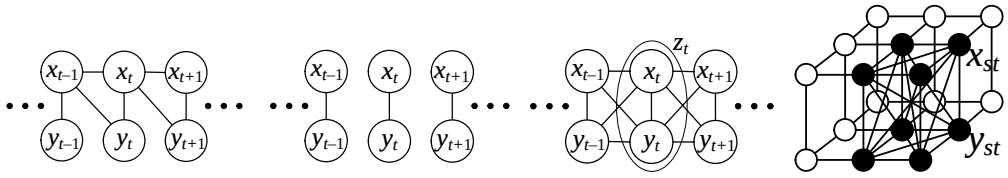
The use of mutual information is well established in this context, as it is theoretically well justified. Unfortunately, it typically involves density estimation or at least a nontrivial optimization procedure [12]. This problem can be averted by using the Hilbert Schmidt Independence Criterion (HSIC). The latter enjoys concentration of measure properties and it can be computed efficiently on any domain where a Reproducing Kernel Hilbert Space (RKHS) can be defined.

However, the application of HSIC is limited to independent and identically distributed (*iid*) data, a property that many problems do not share (*e.g.*, BSS on audio data). For instance many random variables have a pronounced temporal or spatial structure. A simple motivating example is given in Figure 1a. Assume that the observations x_t are drawn *iid* from a uniform distribution on $\{0, 1\}$ and y_t is determined by an XOR operation via $y_t = x_t \otimes x_{t-1}$. Algorithms which treat the observation pairs $\{(x_t, y_t)\}_{t=1}^{\infty}$ as *iid* will consider the random variables x, y as independent. However, it is trivial to detect the XOR dependence by using the information that x_i and y_i are, in fact, sequences.

In view of its importance, temporal correlation has been exploited in the independence test for blind source separation. For instance, [9] used this insight to reject nontrivial nonseparability of nonlinear mixtures, and [19] exploited multiple time-lagged second-order correlations to decorrelate over time.

*This is the long version. A short version published at NIPS is available at http://nips.cc.

†This work was partially done when the author was with the Statistical Machine Learning Group of NICTA.



(a) XOR sequence (b) *iid* (c) First order sequential (d) 2-Dim mesh

Figure 1: From left to right: (a) Graphical model representing the XOR sequence, (b) a graphical model representing *iid* observations, (c) a graphical model for first order sequential data, and (d) a graphical model for dependency on a two dimensional mesh.

These methods work well in practice. But they are rather *ad hoc* and appear very different from standard criteria. In this paper, we propose a framework which extends HSIC to structured non-*iid* data. Our new approach is built upon the connection between exponential family models and the marginal polytope in an RKHS. This is doubly attractive since distributions can be uniquely identified by the expectation operator in the RKHS and moreover, for distributions with conditional independence properties the expectation operator decomposes according to the clique structure of the underlying undirected graphical model [2].

2 The Problem

Denote by $\mathcal{X}$ and $\mathcal{Y}$ domains from which we will be drawing observations $Z := \{(x_1, y_1), \ldots, (x_m, y_m)\}$ according to some distribution $p(x, y)$ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Note that the domains $\mathcal{X}$ and $\mathcal{Y}$ are fully general and we will discuss a number of different structural assumptions on them in Section 3 which allow us to recover existing and propose new measures of dependence. For instance x and y may represent sequences or a mesh for which we wish to establish dependence.

To assess whether x and y are independent we briefly review the notion of Hilbert Space embeddings of distributions [6]. Subsequently we discuss properties of the expectation operator in the case of conditionally independent random variables which will lead to a template for a dependence measure.

Hilbert Space Embedding of Distribution Let $\mathcal{H}$ be a RKHS on $\mathcal{Z}$ with kernel $v : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}$. Moreover, let $\mathcal{P}$ be the space of all distributions over $\mathcal{Z}$, and let $p \in \mathcal{P}$. The expectation operator in $\mathcal{H}$ and its corresponding empirical average can be defined as in [6]

$$\mu[p] := \mathbf{E}_{z \sim p(z)} [v(z, \cdot)] \quad \text{such that} \quad \mathbf{E}_{z \sim p(z)}[f(z)] = \langle \mu[p], f \rangle \tag{1}$$

$$\mu[Z] := \frac{1}{m} \sum_{i=1}^{m} v((x^i, y^i), \cdot) \quad \text{such that} \quad \frac{1}{m} \sum_{i=1}^{m} f(x^i, y^i) = \langle \mu[Z], f \rangle . \tag{2}$$

The map $\mu : \mathcal{P} \mapsto \mathcal{H}$ characterizes a distribution by an element in the RKHS. The following theorem shows that the map is injective [17] for a large class of kernels such as Gaussian and Laplacian RBF.

Theorem 1 *If $\mathbf{E}_{z \sim p} [v(z, z)] < \infty$ and $\mathcal{H}$ is dense in the space of bounded continuous functions $C^0(\mathcal{Z})$ in the L_∞ norm then the map μ is injective.*

2.1 Exponential Families

We are interested in the properties of $\mu[p]$ in the case where p satisfies the conditional independence relations specified by an undirected graphical model. In [2], it is shown that for this case the sufficient statistics decompose along the maximal cliques of the conditional independence graph.

More formally, denote by $\mathcal{C}$ the set of maximal cliques of the graph G and let z_c be the restriction of $z \in \mathcal{Z}$ to the variables on clique $c \in \mathcal{C}$. Moreover, let v_c be universal kernels in the sense of [18] acting on the restrictions of $\mathcal{Z}$ on clique $c \in \mathcal{C}$. In this case, [2] showed that

$$v(z, z') = \sum_{c \in \mathcal{C}} v_c(z_c, z'_c) \tag{3}$$

can be used to describe all probability distributions with the above mentioned conditional independence relations using an exponential family model with v as its kernel. Since for exponential families expectations of the sufficient statistics yield injections, we have the following result:

Corollary 2 *On the class of probability distributions satisfying conditional independence properties according to a graph G with maximal clique set $\mathcal{C}$ and with full support on their domain, the operator*

$$\mu[p] = \sum_{c \in \mathcal{C}} \mu_c[p_c] = \sum_{c \in \mathcal{C}} \mathbf{E}_{z_c} \left[v_c(z_c, \cdot) \right] \tag{4}$$

is injective if the kernels v_c are all universal. The same decomposition holds for the empirical counterpart $\mu[Z]$.

The condition of full support arises from the conditions of the Hammersley-Clifford Theorem [4, 8]: without it, not all conditionally independent random variables can be represented as the product of potential functions. Corollary 2 implies that we will be able to perform all subsequent operations on structured domains simply by dealing with mean operators on the corresponding maximal cliques.

2.2 Hilbert Schmidt Independence Criterion

Theorem 1 implies that we can quantify the difference between two distributions p and q by simply computing the square distance between their RKHS embeddings, *i.e.*, $\|\mu[p] - \mu[q]\|^2_{\mathcal{H}}$. Similarly, we can quantify the strength of dependence between random variables x and y by simply measuring the square distance between the RKHS embeddings of the joint distribution $p(x, y)$ and the product of the marginals $p(x) \cdot p(y)$ via

$$I(x, y) := \|\mu[p(x, y)] - \mu[p(x)p(y)]\|^2_{\mathcal{H}}. \tag{5}$$

Moreover, Corollary 2 implies that for an exponential family consistent with the conditional independence graph G we may decompose $I(x, y)$ further into

$$\begin{aligned} I(x, y) &= \sum\nolimits_{c \in \mathcal{C}} \|\mu_c[p_c(x_c, y_c)] - \mu_c[p_c(x_c)p_c(y_c)]\|^2_{\mathcal{H}_c} \\ &= \sum\nolimits_{c \in \mathcal{C}} \left\{ \mathbf{E}_{(x_c y_c)(x'_c y'_c)} + \mathbf{E}_{x_c y_c x'_c y'_c} - 2\mathbf{E}_{(x_c y_c) x'_c y'_c} \right\} \left[v_c((x_c, y_c), (x'_c, y'_c)) \right] \end{aligned} \tag{6}$$

where bracketed random variables in the subscripts are drawn from their joint distributions and unbracketed ones are from their respective marginals, *e.g.*, $\mathbf{E}_{(x_c y_c) x'_c y'_c} := \mathbf{E}_{(x_c y_c)} \mathbf{E}_{x'_c} \mathbf{E}_{y'_c}$. Obviously the challenge is to find good empirical estimates of (6). In its simplest form we may replace each of the expectations by sums over samples, that is, by replacing

$$\mathbf{E}_{(x,y)}[f(x, y)] \leftarrow \frac{1}{m} \sum_{i=1}^{m} f(x_i, y_i) \quad \text{and} \quad \mathbf{E}_{(x)(y)}[f(x, y)] \leftarrow \frac{1}{m^2} \sum_{i,j=1}^{m} f(x_i, y_j). \tag{7}$$

3 Estimates for Special Structures

To illustrate the versatility of our approach we apply our model to a number of graphical models ranging from independent random variables to meshes proceeding according to the following recipe:

1. Define a conditional independence graph.
2. Identify the maximal cliques.
3. Choose suitable joint kernels on the maximal cliques.
4. Exploit stationarity (if existent) in $I(x, y)$ in (6).
5. Derive the corresponding empirical estimators for each clique, and hence for all of $I(x, y)$.

3.1 Independent and Identically Distributed Data

As the simplest case, we first consider the graphical model in Figure 1b, where $\{(x_t, y_t)\}_{t=1}^T$ are *iid* random variables. Correspondingly the maximal cliques are $\{(x_t, y_t)\}_{t=1}^T$. We choose the joint kernel on the cliques to be

$$v_t((x_t, y_t), (x'_t, y'_t)) := k(x_t, x'_t) l(y_t, y'_t) \quad \text{hence } v((x, y), (x', y')) = \sum_{t=1}^{T} k(x_t, x'_t) l(y_t, y'_t). \tag{8}$$

The representation for v_t implies that we are taking an outer product between the Hilbert Spaces on x_t and y_t induced by kernels k and l respectively. If the pairs of random variables (x_t, y_t) are *not* identically distributed, all that is left is to use (8) to obtain an empirical estimate via (7).

We may improve the estimate considerably if we are able to assume that all pairs (x_t, y_t) are drawn from the *same* distribution $p(x_t, y_t)$. Consequently all coordinates of the mean map are identical and we can use all the data to estimate just one of the discrepancies $\|\mu_c[p_c(x_c, y_c)] - \mu_c[p_c(x_c)p_c(y_c)]\|^2$. The latter expression is identical to the standard HSIC criterion and we obtain the biased estimate

$$\hat{I}(x, y) = \tfrac{1}{T} \operatorname{tr} HKHL \quad \text{where} \quad K_{st} := k(x_s, x_t), L_{st} := l(y_s, y_t) \text{ and } H_{st} := \delta_{st} - \tfrac{1}{T}. \quad (9)$$

3.2 Sequence Data

A more interesting application beyond *iid* data is sequences with a Markovian dependence as depicted in Figure 1c. Here the maximal cliques are the sets $\{(x_t, x_{t+1}, y_t, y_{t+1})\}_{t=1}^{T-1}$. More generally, for longer range dependency of order $\tau \in \mathbb{N}$, the maximal cliques will involve the random variables $(x_t, \ldots, x_{t+\tau}, y_t, \ldots, y_{t+\tau}) =: (x_{t,\tau}, y_{t,\tau})$.

We assume homogeneity and stationarity of the random variables: that is, all cliques share the same sufficient statistics (feature map) and their expected value is identical. In this case the kernel

$$v_0((x_{t,\tau}, y_{t,\tau}), (x'_{t,\tau}, y'_{t,\tau})) := k(x_{t,\tau}, x'_{t,\tau}) l(y_{t,\tau}, y'_{t,\tau})$$

can be used to measure discrepancy between the random variables. Stationarity means that $\mu_c[p_c(x_c, y_c)]$ and $\mu_c[p_c(x_c)p_c(y_c)]$ are the same for all cliques c, hence $I(x, y)$ is a multiple of the difference for a single clique.

Using the same argument as in the *iid* case, we can obtain a biased estimate of the measure of dependence by using $K_{ij} = k(x_{i,\tau}, x_{j,\tau})$ and $L_{ij} = l(y_{i,\tau}, y_{j,\tau})$ instead of the definitions of K and L in (9). This works well in experiments. In order to obtain an unbiased estimate we need some more work. Recall the unbiased estimate of $I(x, y)$ is a fourth order U-statistic [6].

Theorem 3 *An unbiased empirical estimator for* $\|\mu[p(x, y)] - \mu[p(x)p(y)]\|^2$ *is*

$$\hat{I}(x, y) := \tfrac{(m-4)!}{m!} \sum_{(i,j,q,r)} h(x_i, y_i, \ldots, x_r, y_r), \quad (10)$$

where the sum is over all terms such that i, j, q, r *are mutually different, and*

$$h(x_1, y_1, \ldots, x_4, y_4) := \frac{1}{4!} \sum_{(t,u,v,w)}^{(1,2,3,4)} k(x_t, x_u) l(x_t, x_u) + k(x_t, x_u) l(x_v, x_w) - 2k(x_t, x_u) l(x_t, x_v),$$

and the latter sum denotes all ordered quadruples (t, u, v, w) *drawn from* $(1, 2, 3, 4)$.

The theorem implies that in expectation h takes on the value of the dependence measure. To establish that this also holds for dependent random variables we use a result from [1] which establishes convergence for stationary mixing sequences under mild regularity conditions, namely whenever the kernel of the U-statistic h is bounded and the process generating the observations is absolutely regular. See also [5, Section 4].

Theorem 4 *Whenever* $I(x, y) > 0$*, that is, whenever the random variables* are *dependent, the estimate* $\hat{I}(x, y)$ *is asymptotically normal with*

$$\sqrt{m}(\hat{I}(x, y) - I(x, y)) \xrightarrow{d} \mathcal{N}(0, 4\sigma^2) \quad (11)$$

where the variance is given by

$$\sigma^2 = \operatorname{Var}[h_3(x_1, y_1)]^2 + 2 \sum_{t=1}^{\infty} \operatorname{Cov}(h_3(x_1, y_1), h_3(x_t, y_t)) \quad (12)$$

$$\text{and} \quad h_3(x_1, y_1) := \mathbf{E}_{(x_2, y_2, x_3, y_3, x_4, y_4)}[h(x_1, y_1, \ldots, x_4, y_4)] \quad (13)$$

This follows from [5, Theorem 7], again under mild regularity conditions (note that [5] state their results for U-statistics of second order, and claim the results hold for higher orders). The proof is tedious but does not require additional techniques and is therefore omitted.

3.3 TD-SEP as a special case

So far we did not discuss the freedom of choosing different kernels. In general, an RBF kernel will lead to an effective criterion for measuring the dependence between random variables, especially in time-series applications. However, we could also choose linear kernels for k and l, for instance, to obtain computational savings.

For a specific choice of cliques and kernels, we can recover the work of [19] as a special case of our framework. In [19], for two centered scalar time series x and y, the contrast function is chosen as the sum of same-time and time-lagged cross-covariance $\mathbf{E}[x_t y_t] + \mathbf{E}[x_t y_{t+\tau}]$. Using our framework, two types of cliques, (x_t, y_t) and $(x_t, y_{t+\tau})$, are considered in the corresponding graphical model. Furthermore, we use a joint kernel of the form

$$\langle x_s, x_t \rangle \langle y_s, y_t \rangle + \langle x_s, x_t \rangle \langle y_{s+\tau}, y_{t+\tau} \rangle , \tag{14}$$

which leads to the estimator of structured HSIC: $I(x, y) = \frac{1}{T} (\operatorname{tr} HKHL + \operatorname{tr} HKHL_\tau)$. Here L_τ denotes the linear covariance matrix for the time lagged y signals. For scalar time series, basic algebra shows that $\operatorname{tr} HKHL$ and $\operatorname{tr} HKHL_\tau$ are the estimators of $\mathbf{E}[x_t y_t]$ and $\mathbf{E}[x_t y_{t+\tau}]$ respectively (up to a multiplicative constant).

Further generalization can incorporate several time lagged cross-covariances into the contrast function. For instance, TD-SEP [19] uses a range of time lags from 1 to τ. That said, by using a nonlinear kernel we are able to obtain better contrast functions, as we will show in our experiments.

3.4 Grid Structured Data

Structured HSIC can go beyond sequence data and be applied to more general dependence structures such as 2-D grids for images. Figure 1d shows the corresponding graphical model. Here each node of the graphical model is indexed by two subscripts, i for row and j for column. In the simplest case, the maximal cliques are

$$\mathcal{C} = \{(x_{ij}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1}, y_{ij}, y_{i+1,j}, y_{i,j+1}, y_{i+1,j+1})\}_{ij}.$$

In other words, we are using a cross-shaped stencil to connect vertices. Provided that the kernel v can also be decomposed into the product of k and l, then a biased estimate of the independence measure can be again formulated as $\operatorname{tr} HKHL$ up to a multiplicative constant. The statistical analysis of U-statistics for stationary Markov random fields is highly nontrivial. We are not aware of results equivalent to those discussed in Section 3.2.

4 Experiments

Having a dependence measure for structured spaces is useful for a range of applications. Analogous to *iid* HSIC, structured HSIC can be applied to non-*iid* data in applications such as independent component analysis [13], independence test [6], feature selection [16], clustering [15], and dimensionality reduction [14]. The fact that structured HSIC can take into account the interdependency between observations provides us with a principled generalization of these algorithms to, *e.g.*, time series analysis. In this paper, we will focus on three examples: 1. independence test where structured HSIC is used as a test statistic, 2. independent component analysis where we wish to minimize the dependence, and 3. time series segmentation where we wish to maximize the dependence instead.

4.1 Independence Test

We first present two experiments that use the structured HSIC as an independence measure for non-*iid* data, namely XOR binary sequence and Gaussian process. With structured HSIC as a test statistic, we still need an approach to building up the distribution of the test statistic under the null hypothesis H0 : $x \perp\!\!\!\perp y$. For this purpose, we generalize the random shuffling technique commonly used for *iid* observations [6] into a clique-bundled shuffling. This shuffling technique randomly pairs up the observations in x and y. Depending on the clique configurations of structured HSIC, one observation in x may be paired up with several observations in y. The observations corresponding to an instance of a maximal clique need to be bundled together and shuffled in blocks. For instance, if the maximal cliques are $\{(x_t, y_t, y_{t+1})\}$, after shuffling we may have pairs such as (x_3, y_8, y_9) and (x_8, y_3, y_4), but never have pairs such as (x_3, y_4, y_9) or (x_4, y_3, y_8), because y_3 is bundled with y_4, and y_8 is bundled with y_9. If structured HSIC has a form of (9) with kernels K and L

Table 1: The number of times HSIC and structured HSIC rejected the null hypothesis.

data	HSIC	p-value	Structured HSIC	p-value
XOR	1	0.44±0.29	100	0±0
RAND	1	0.49±0.28	0	0.49±0.31

possibly assuming more general forms like $k(x_{i,\tau}, x_{j,\tau})$, the shuffling can be performed directly on the kernel entries. In this case, the kernel matrices K and L for x and y can be computed offline and separately. Given a permutation π, a shuffle will change L_{st} into $L_{\pi(s)\pi(t)}$. The random shuffling is usually carried out many times and structured HSIC is computed at each time, which results in the null distribution.

4.1.1 Independence Test for XOR Binary Sequences

In this experiment, we compared *iid* HSIC and structured HSIC in terms of their performance on independence test. We generated two binary sequences x and y of length $T = 400$. The observations in x were drawn *iid* from a uniform distribution over $\{0, 1\}$. y were determined by an XOR operation over observations from x: $y_t = x_t \otimes x_{t-1}$. If we treat the observation pairs as *iid*, then the two sequences must appear independent. The undirected graphical model for this data is shown in Figure 1b.

For *iid* HSIC, we used maximal cliques $\{(x_t, y_t)\}$ to reflect its underlying *iid* assumption. The corresponding kernel is $\delta(x_s, x_t)\delta(y_s, y_t)$. The maximal cliques for structured HSIC are $\{(x_{t-1}, x_t, y_t)\}$, which takes into account the interdependent nature of the observations. The corresponding kernel is $\delta(x_{s-1}, x_{t-1})\delta(x_s, x_t)\delta(y_s, y_t)$. We tested the null hypothesis $\mathrm{H0} : x \perp\!\!\!\perp y$ with both methods at significance level 0.01. The distributions of the test statistics was built by shuffling the paring of kernel entries for 1000 times.

We randomly instantiated the two sequences for 100 times, then counted the number of times each method rejected the null hypothesis (Table 1 XOR row). Structured HSIC did a perfect job in detecting the dependence between the sequences, while normal HSIC almost completely missed that out. For comparison, we also generated a second dataset with two independent and uniformly distributed binary sequences. Now both methods correctly detected the independence (Table 1 RAND row). We also report the mean and standard deviation of the p-values over the 100 instantiations of the experiment to give a rough picture of the distribution of the p-values.

4.1.2 Independence Test for Gaussian Processes

In this experiment, we generated two sequences $x = \{x_t\}_{t=1}^T$ and $y = \{y_t\}_{t=1}^T$ using the following formulae:

$$x = Au \quad \text{and} \quad y = A\left(\varepsilon u + \sqrt{1-\varepsilon^2}v\right), \tag{15}$$

where $A \in \mathbb{R}^{T\times T}$ is a mixing matrix, and $u = \{u_t\}_{t=1}^T$ and $v = \{v_t\}_{t=1}^T$ are sequences of *iid* zero-mean and unit-variance normal observations. $\varepsilon \in [0, 1]$ and larger values of ε lead to higher dependence between sequences x and y. In this setting, both x and y are stationary Gaussian processes. Furthermore, due to the mixing matrix A (especially its non-zero off-diagonal elements), observations within x and y are interdependent. We expect that an independence test which takes into account this structure will outperform tests assuming *iid* observations. In our experiment, we used $T = 2000$ and $A_{ab} = \exp(-|a-b|/25)$ with all elements below 0.7 clamped to 0. This banded matrix makes the interdependence in x and y localized. For structured HSIC, we used the maximal cliques $\{(x_{t,\tau}, y_{t,\tau})\}$ where $\tau = 10$ and linear kernel $\langle x_{s,10}, x_{t,10}\rangle \langle y_{s,10}, y_{t,10}\rangle$.

We varied $\varepsilon \in \{0, 0.05, 0.1, \ldots, 0.7\}$. For each value of ε, we randomly instantiated u and v for 1000 times. For each instantiation, we followed the strategy in [10] which formed a new subsequence of length 200 by resampling every d observations and here we used $d = 5$. We tested the null hypothesis $\mathrm{H0} : x \perp\!\!\!\perp y$ with 500 random shuffles, and the nominal risk level was set to $\alpha = 0.01$. When $\varepsilon = 0$ we are interested in the Type I error, *i.e.*, the fraction of times when H0 is rejected which should be no greater than the α. When $\varepsilon > 0$ we are concerned about the same fraction, but now called *empirical power* of the test because a higher value is favored. d and τ

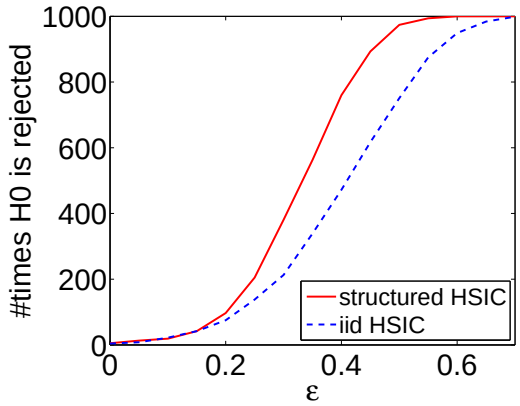


Figure 2: Independence test for Gaussian process.

were chosen to make the comparison fair. Smaller d includes more autocorrelation and increases the empirical power for both *iid* HSIC and structured HSIC, but it causes higher Type I error [see *e.g.*, Table II in 10]. We chose $d = 5$ since it is the smallest d such that Type I error is close to the nominal risk level $\alpha = 0.01$. τ is only for structured HSIC, and in our experiment higher values of τ did not significantly improve the empirical power, but just make the kernels more expensive to compute.

In Figure 2, we plot the number of times H0 is rejected. When $\varepsilon = 0$, x and y are independent and both *iid* HSIC and structured HSIC almost always accept H0. When $\varepsilon \in [0.05, 0.2]$, *i.e.*, x and y are slightly dependent, both tests have a low empirical power. When $\varepsilon > 0.2$, structured HSIC is considerably more sensitive in detecting dependency and consistently rejects H0 more frequently. Note u and v have the same weight in (15) when $\varepsilon = 2^{-1/2} = 0.71$.

4.2 Independent Component Analysis

In independent component analysis (ICA), we observe a time series of vectors u that corresponds to a linear mixture $u = As$ of n mutually independent sources s (each entry in the source vector here is a random process, and depends on its past values; examples include music and EEG time series). Based on the series of observations t, we wish to recover the sources using only the independence assumption on s. Note that sources can only be recovered up to scaling and permutation. The core of ICA is a contrast function that measures the independence of the estimated sources. An ICA algorithm searches over the space of mixing matrix A such that this contrast function is minimized. Thus, we propose to use structured HSIC as the contrast function for ICA. By incorporating time lagged variables in the cliques, we expect that structured HSIC can better deal with the non-*iid* nature of time series. In this respect, we generalize the TD-SEP algorithm [19], which implements this idea using a linear kernel on the signal. Thus, we address the question of whether correlations between higher order moments, as encoded using non-linear kernels, can improve the performance of TD-SEP on real data.

Data Following the setting of [7, Section 5.5], we unmixed various musical sources, combined using a randomly generated orthogonal matrix A (since optimization over the orthogonal part of a general mixing matrix is the more difficult step in ICA). We considered mixtures of two to four sources, drawn at random without replacement from 17 possibilities. We used the sum of pairwise dependencies as the overall contrast function when more than two sources were present.

Methods We compared structured HSIC to TD-SEP and *iid* HSIC. While *iid* HSIC does not take the temporal dependence in the signal into account, it has been shown to perform very well for *iid* data [13]. Following [7], we employed a Laplace kernel, $k(x, x') = \exp(-\lambda\|x - x'\|)$ with $\lambda = 3$ for both structured and *iid* HSIC. For both structured and *iid* HSIC, we used gradient descent over the orthogonal group with a Golden search, and low rank Cholesky decompositions of the Gram matrices to reduce computational cost, as in [3].

Table 2: Median performance of ICA on music using HSIC, TDSEP, and structured HSIC. In the top row, the number n of sources and m of samples are given. In the second row, the number of time lags τ used by TDSEP and structured HSIC are given: thus the observation vectors $x, x_{t-1}, \ldots, x_{t-\tau}$ were compared. The remaining rows contain the median Amari divergence (multiplied by 100) for the three methods tested. The original HSIC method does not take into account time dependence ($\tau = 0$), and returns a single performance number. Results are in all cases averaged over 136 repetitions: for two sources, this represents all possible pairings, whereas for larger n the sources are chosen at random without replacement.

Method	$n = 2, m = 5000$			$n = 3, m = 10000$			$n = 4, m = 10000$		
	1	2	3	1	2	3	1	2	3
HSIC	1.51			1.70			2.68		
TDSEP	1.54	1.62	1.74	1.84	1.72	1.54	2.90	2.08	1.91
Structured HSIC	1.48	1.62	1.64	1.65	1.58	1.56	2.65	2.12	1.83

Results We chose the Amari divergence as the index for comparing performance of the various ICA methods. This is a divergence measure between the estimated and true unmixing matrices, which is invariant to the output ordering and scaling ambiguities. A smaller Amari divergence indicates better performance. Results are shown in Table 2. Overall, contrast functions that take time delayed information into account perform best, although the best time lag is different when the number of sources varies.

4.3 Time Series Clustering and Segmentation

We can also extend clustering to time series and sequences using structured HSIC. This is carried out in a similar way to the *iid* case. One can formulate clustering as generating the labels y from a finite discrete set, such that their dependence on x is maximized [15]:

$$\text{maximize}_y \;\; \operatorname{tr} HKHL \quad \textit{subject to} \text{ constraints on } y. \tag{16}$$

Here K and L are the kernel matrices for x and the generated y respectively. More specifically, assuming $L_{st} := \delta(y_s, y_t)$ for discrete labels y, we recover clustering. Relaxing discrete labels to $y_t \in \mathbb{R}$ with bounded norm $\|y\|_2$ and setting $L_{st} := y_s y_t$, we obtain Principal Component Analysis.

This reasoning for *iid* data carries over to sequences by introducing additional dependence structure through the kernels: $K_{st} := k(x_{s,\tau}, x_{t,\tau})$ and $L_{st} := l(y_{s,\tau}, y_{t,\tau})$. In general, the interacting label sequences make the optimization in (16) intractable. However, for a class of kernels l an efficient decomposition can be found by applying a reverse convolution on k.

4.3.1 Efficient Optimization for Convolution Kernels

Suppose the kernel l assumes a special form given by

$$l(y_{s,\tau}, y_{t,\tau}) = \sum\nolimits_{u,v=0}^{\tau} \bar{l}(y_{s+u}, y_{t+v}) M_{uv}, \tag{17}$$

where $M \in \mathbb{R}^{(\tau+1)\times(\tau+1)}$ is positive semi-definite, and $\bar{l}$ is a base kernel between individual time points. A common choice is $\bar{l}(y_s, y_t) = \delta(y_s, y_t)$. In this case we can rewrite $\operatorname{tr} HKHL$ by applying the summation over M to HKH, *i.e.*,

$$\sum_{s,t=1}^{T} [HKH]_{ij} \sum_{u,v=0}^{\tau} \bar{l}(y_{s+u}, y_{t+v}) M_{uv} = \sum_{s,t=1}^{T+\tau} \underbrace{\sum_{\substack{u,v=0 \\ s-u,t-v\in[1,T]}}^{\tau} M_{uv} [HKH]_{s-u,t-v}}_{:=\bar{K}_{st}} \bar{l}(y_s, y_t) \tag{18}$$

This means that we may apply the matrix M to HKH and thereby we are able to decouple the dependency within y. That is, in contrast to l which couples two subsequences of y, $\bar{l}$ only couples two individual elements of y. As a result, the optimization over y is made much easier. Denoting the convolution by $\bar{K} = [HKH] \star M$, we can directly apply (16) to time series and sequence data in the same way as *iid* data, treating $\bar{K}$ as the original K. In practice, approximate algorithms such as incomplete Cholesky decomposition are needed to efficiently compute and represent $\bar{K}$.

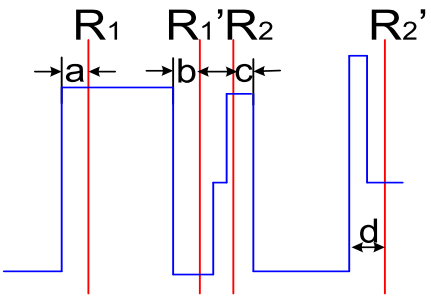


Figure 3: Illustration of error calculation. Red lines denote the ground truth and blues line are the segmentation results. The error introduced for segment R_1 to R_1' is $a+b$, while that for segment R_2 to R_2' is $c+d$. The overall error in this example is then $(a+b+c+d)/4$.

4.3.2 Empirical Evaluation

Datasets We studied two datasets in this experiment.

1. **Swimming Dataset**. The first dataset was collected by the Australian Institute of Sports (AIS) from a 3-channel orientation sensor attached to a swimmer which monitors: 1. the body orientation by a 3-channel magnetometer; 2. the acceleration by a 3-channel accelerometer. The three time series we used in our experiment have the following configurations: $T = 23000$ time steps with 4 laps; $T = 47000$ time steps with 16 laps; and $T = 67000$ time steps with 20 laps. The task is to automatically find the starting and finishing time of each lap based on the sensor signals. We treated this problem as a segmentation problem, and used orientation data for our experiments because they lead to better results than the acceleration signals. Since the dataset contains four different styles of swimming, we assumed there are six states/clusters for the sequence: four clusters for the four styles of swim, two clusters for approaching and leaving the end of the pool (finishing and starting a lap, respectively).

2. **BCI dataset**. The second dataset is a brain-computer interface data (data IVb of Berlin BCI group[1]). It contains EEG signals collected when a subject was performing three types of cued imagination: `left`, `foot`, and `relax`. Between every two successive imaginations, there is an interim. So an example state sequence is:

```
left, interim, relax, interim, foot, interim, relax, interim,...
```

Therefore, the `left/foot/relax` states correspond to the swimming styles and the interim corresponds to the turning at the end or beginning of the laps. Including the interim period, the dataset consists of $T = 10000$ time points with 16 different segments (32 boundaries). The task is to automatically detect the start and end of an imagination. We used four clusters for this problem.

We preprocessed the raw signal sequences by applying them to a bandpass filter which only keeps the frequency range from 12Hz to 14Hz. Besides, we followed the common practice and only used the following electrode channels (basically those in the middle of the test region):

```
33,34,35,36,37,38,39,42,43,44,45,46,47,48,49,51,52,53,54,
55,56,57,59,60,61,62,63,64,65,66,69,70,71,72,73,74,75.
```

Finally, for both swimming and BCI datasets, we smoothed the raw data with moving averages, *i.e.*, $x_t \leftarrow \sum_{\tau=-w}^{w} x_{t+\tau}^{\text{raw}}$ followed by normalization to zero mean and unit variance for each feature dimension. Here w is set to 100 for swimming data and 50 for BCI data due to its higher frequency of state switching. This smoothed and normalized x was used by *ALL* the three algorithms.

Methods We compared three algorithms: structured HSIC for clustering, spectral clustering [11], and HMM.

1. **Structured HSIC**. For the three swimming datasets, we used the maximal cliques of $\{(x_t, y_{t-50,100})\}$ for structured HSIC, where y is the discrete label sequence to be generated.

[1] http://ida.first.fraunhofer.de/projects/bci/competition-iii/desc-IVb.html

Table 3: Segmentation errors by various methods on the four studied time series.

Method	Swimming 1	Swimming 2	Swimming 3	BCI
structured HSIC	**99.0**	**118.5**	**108.6**	**111.5**
spectral clustering	125	212.3	143.9	162
HMM	153.2	120	150	168

Time lagged labels in the maximal cliques reflect the fact that clustering labels keep the same for a period of time. The kernel l on y took the form of equation (17), with $M \in \mathbb{R}^{101\times 101}$ and $M_{ab} := \exp(-(a-b)^2)$. We used the technique described in Section 4.3.1 to shift the dependence within y into x. The kernel k on x was RBF: $\exp(-\|x_s - x_t\|^2)$. We performed kernel k-means clustering based on the convolved kernel matrix $\bar{K}$. To avoid the local minima of k-means, we randomly initialized it for 20 times and reported the error made by the model which has the lowest sum of point-to-centroid distances. The parameters for BCI dataset are the same, except that $M \in \mathbb{R}^{51\times 51}$ to reflect the fact that state changes more frequently in this dataset.

2. **Spectral clustering.** We first applied the algorithm in [11] on x and it yielded far larger error, and hence is not reported here. Then we applied its kernelized version to the convolved kernel $\bar{K}$. We used 100 nearest neighbors with distance function $\exp(-\|x_i - x_j\|^2)$. These parameters delivered uniformly best result.

3. **HMM**. We trained a first order homogeneous HMM by the EM algorithm with 6 hidden states for swimming dataset and 4 states for BCI dataset, and its observation model contained diagonal Gaussians. After training, we used Viterbi decoding to determine the cluster labels. We used the implementation from Torch[2]. To regularize, we tried a range of minimum variance $\sigma \in \{0.5, 0.6, ..., 2.0\}$. For each σ, we randomly initialized the training of HMM for 50 times to avoid local maxima of EM, and computed the error incurred by the model which yielded the highest likelihood on the whole sequence. Finally, we reported the minimum error over all σ.

Results To evaluate the segmentation quality, the boundaries found by various methods were compared against the ground truth. First, each detected boundary was matched to a true boundary, and then the discrepancy between them was counted into the error. The overall error was this sum divided by the number of boundaries. Figure 3 gives an example on how to compute this error.

According to Table 3, in all of the four time series we studied, segmentation using structured HSIC leads to lower error compared with spectral clustering and HMM. For instance, structured HSIC reduces nearly $1/3$ of the segmentation error in the BCI dataset. We also plot the true boundaries together with the segmentation results produced by structured HSIC, spectral clustering, and HMM respectively. Figures 5 to 7 present the results for the three swimming datasets, and Figure 4 for the BCI dataset. Although the results of swimming data in Figure 5 to 7 are visually similar among all algorithms, the average error produced by structured HSIC is much smaller than that of HMM or spectral clustering. Finally, the segment boundaries of BCI data produced by structured HSIC clearly fit better with the ground truth.

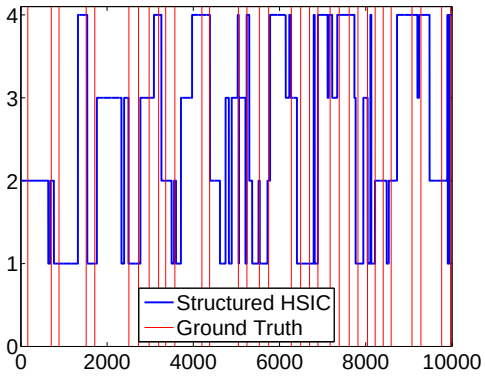
5 Conclusion

In this paper, we extended the Hilbert Schmidt Independence Criterion from *iid* data to structured and non-*iid* data. Our approach is based on RKHS embeddings of distributions, and utilizes the efficient factorizations provided by the exponential family associated with undirected graphical models. Encouraging experimental results were demonstrated on independence test, ICA, and segmentation for time series. Further work will be done in the direction of applying structured HSIC to PCA and feature selection on structured data.

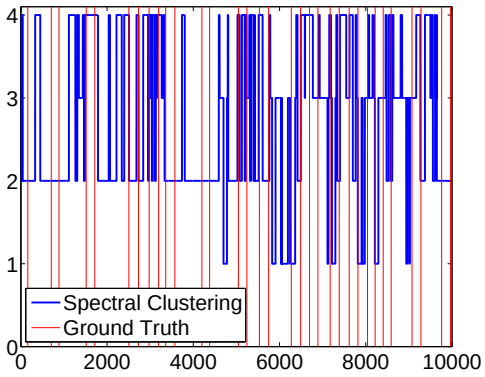
Acknowledgements

NICTA is funded by the Australian Governments Backing Australias Ability and the Centre of Excellence programs. This work is also supported by the IST Program of the European Community, under the FP7 Network of Excellence, ICT-216886-NOE.

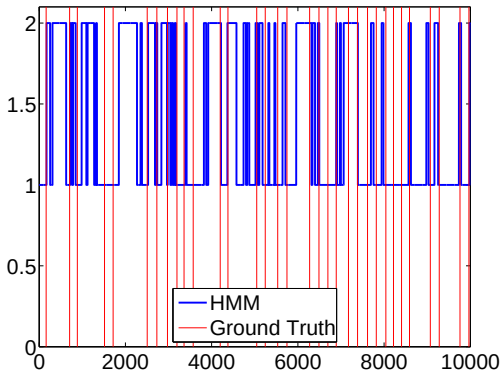
[2] http://www.torch.ch



(a) Structured HSIC

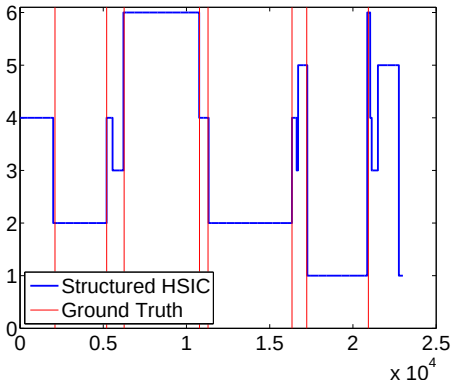


(b) Spectral Clustering

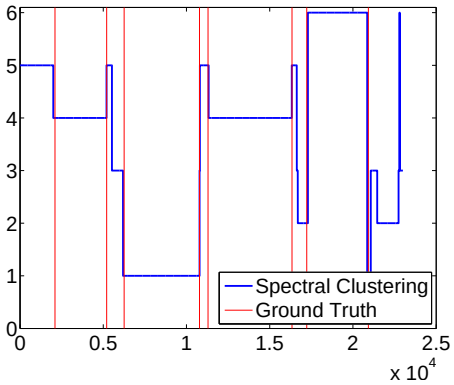


(c) HMM

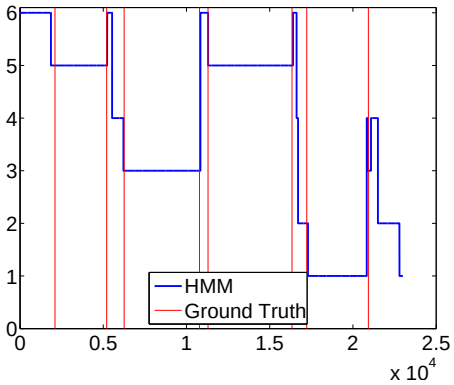
Figure 4: Segmentation results of BCI dataset produced by (a) structured HSIC, (b) spectral clustering and (c) HMM. In (c), we did specify 4 hidden states, but the Viterbi decoding showed only two states were used.



(a) Structured HSIC

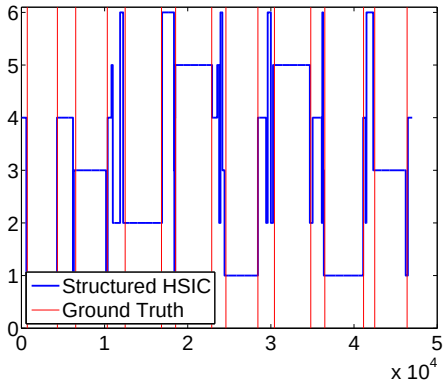


(b) Spectral Clustering

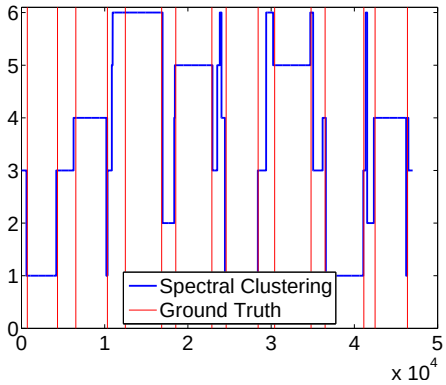


(c) HMM

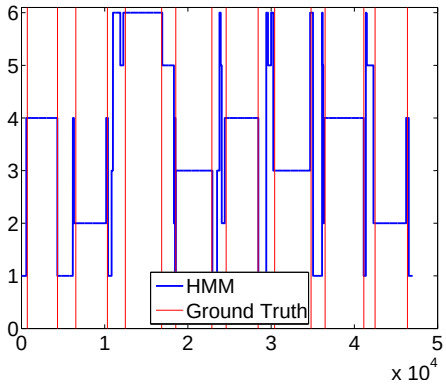
Figure 5: Segmentation results of swimming dataset 1 produced by (a) structured HSIC, (b) spectral clustering and (c) HMM.



(a) Structured HSIC

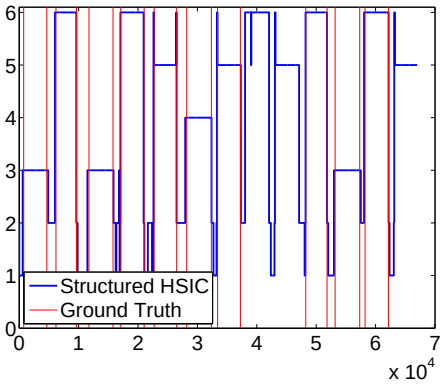


(b) Spectral Clustering

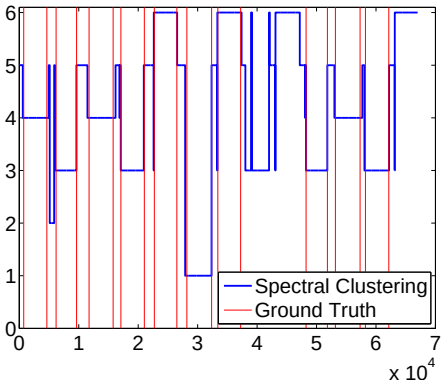


(c) HMM

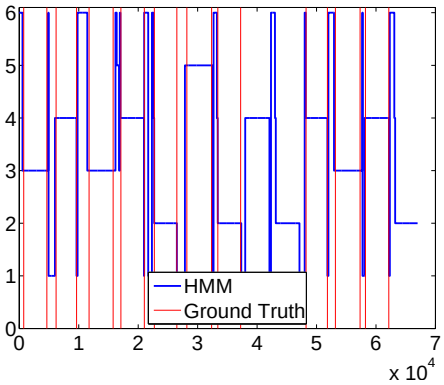
Figure 6: Segmentation results of swimming dataset 2 produced by (a) structured HSIC, (b) spectral clustering and (c) HMM.



(a) Structured HSIC



(b) Spectral Clustering



(c) HMM

Figure 7: Segmentation results of swimming dataset 3 produced by (a) structured HSIC, (b) spectral clustering and (c) HMM.

References

[1] Aaronson, J., Burton, R., Dehling, H., Gilat, D., Hill, T., & Weiss, B. (1996). Strong laws for L and U-statistics. *Transactions of the American Mathematical Society*, *348*, 2845–2865.

[2] Altun, Y., Smola, A. J., & Hofmann, T. (2004). Exponential families for conditional random fields. In *UAI*.

[3] Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *JMLR*, *3*, 1–48.

[4] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Roy. Stat. Soc. B*, *36*(B), 192–326.

[5] Borovkova, S., Burton, R., & Dehling, H. (2001). Limit theorems for functionals of mixing processes with applications to dimension estimation. *Transactions of the American Mathematical Society*, *353*(11), 4261–4318.

[6] Gretton, A., Fukumizu, K., Teo, C.-H., Song, L., Schölkopf, B., & Smola, A. (2008). A kernel statistical test of independence. Tech. Rep. 168, MPI for Biological Cybernetics.

[7] Gretton, A., Herbrich, R., Smola, A., Bousquet, O., & Schölkopf, B. (2005). Kernel methods for measuring independence. *JMLR*, *6*, 2075–2129.

[8] Hammersley, J. M., & Clifford, P. E. (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

[9] Hosseni, S., & Jutten, C. (2003). On the separability of nonlinear mixtures of temporally correlated sources. *IEEE Signal Processing Letters*, *10*(2), 43–46.

[10] Karvanen, J. (2005). A resampling test for the total independence of stationary time series: Application to the performance evaluation of ICA algorithms. *Neural Processing Letters*, *22*(3), 311 – 324.

[11] Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *NIPS*.

[12] Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2008). Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *NIPS*.

[13] Shen, H., Jegelka, S., & Gretton, A. (submitted). Fast kernel-based independent component analysis. *IEEE Transactions on Signal Processing*.

[14] Song, L., Smola, A., Borgwardt, K., & Gretton, A. (2007). Colored maximum variance unfolding. In *NIPS*.

[15] Song, L., Smola, A., Gretton, A., & Borgwardt, K. (2007). A dependence maximization view of clustering. In *Proc. Intl. Conf. Machine Learning*.

[16] Song, L., Smola, A., Gretton, A., Borgwardt, K., & Bedo, J. (2007). Supervised feature selection via dependence estimation. In *ICML*.

[17] Sriperumbudur, B., Gretton, A., Fukumizu, K., Lanckriet, G., & Schölkopf, B. (2008). Injective hilbert space embeddings of probability measures. In *COLT*.

[18] Steinwart, I. (2002). The influence of the kernel on the consistency of support vector machines. *JMLR*, *2*.

[19] Ziehe, A., & Müller, K.-R. (1998). TDSEP – an efficient algorithm for blind separation using time structure. In *ICANN*.