ON DATA SPARSIFICATION AND A RECURSIVE ALGORITHM FOR ESTIMATING A KERNEL-BASED MEASURE OF INDEPENDENCE

Pierre-Olivier Amblard[1,2]

[1] Dept. of Mathematics&Statistics
The University of Melbourne, Australia
[2] GIPSA-lab, UMR CNRS 5216
Grenoble, France

Jonathan H. Manton[3]

[3] Control and Signal Processing Lab
The University of Melbourne, Australia

ABSTRACT

Technological improvements have led to situations where data sets are sufficiently rich that in the interests of processing speed it is desirable to throw away samples that provide little additional information. This is referred to here as data sparsification. The first contribution is a study of a recently proposed data sparsification scheme; ideas from vector quantisation are used to assess its performance. Informed by this study, a modification of the data sparsification algorithm is proposed and applied to the problem of estimating a kernel-based measure of independence of two datasets. (Given *i.i.d.* observations from two random variables, x and y, the underlying problem is to determine whether or not x and y are independent of each other.) The second contribution of this paper is to make recursive an existing algorithm for measuring independence and able to operate on both raw data and on sparsified data generated by the aforementioned data sparsification algorithm. Compared with the original algorithm, the recursive algorithm is significantly faster due to its lower memory and computational requirements.

Index Terms— sparse, dictionary, kernel, quantisation, independence

1. INTRODUCTION

Roughly speaking, kernel-based methods work by converting low-dimensional nonlinear problems into high-dimensional linear problems. Importantly, by choosing the high-dimensional space to be a reproducing kernel Hilbert space (RKHS), geometric operations such as evaluating inner products can be computed efficiently. Nevertheless, as the number of data points increases, the corresponding Gram matrix (defined in Section 3) becomes larger and more costly to work with. There is therefore an incentive to reduce the raw number of data points fed into a kernel-based algorithm.

Of particular interest in this paper is a kernel-based algorithm for computing an empirical measure of independence. Such an algorithm takes as input pairs of i.i.d. observations $\{(x_i, y_i\}$, $i = 1, 2, \cdots, N\}$, and outputs a nonnegative number whose closeness to zero is a measure of independence between the random variables X and Y. (In fact, this number is the Hilbert-Schmidt norm of a certain linear operator constructed from the data and, under mild conditions, will approach zero as the number of data points grows if and only if X and Y are independent.) The two main contributions of the paper are:

1. the derivation of a *recursive* algorithm for computing an empirical measure of independence of pairwise observations; and
2. a data sparsification front-end for the aforementioned algorithm that (greatly) reduces the number of pairwise observations required for an accurate measure of independence.

The second contribution is interesting in its own right because a wide range of algorithms can benefit from data sparsification. The underlying idea is straightforward: given i.i.d. observations $Z_1, Z_2, \cdots$, for most intents and purposes, it suffices to form an empirical histogram of the data and replace the original data by data resampled from this empirical histogram. Essentially the only question here is an appropriate choice of bin size used to form the histogram.

One way of doing this is to build up a dictionary of observations. As each new data point is observed, it is compared with all the existing entries in the dictionary. If it is sufficiently close to a dictionary entry, the data point is replaced by the closest dictionary entry; in Section 2, this is likened to data quantisation. Otherwise, the new observation is left unchanged and added to the dictionary. This can be viewed as forming an empirical histogram using adaptive bin sizes. Of course, the issue remains of how to measure the closeness of a new observation to those in the dictionary. In the machine-learning literature, a so-called coherence criterion is commonly used. This is discussed in Section 2, along with several extensions of existing data sparsification algorithms.

Section 3 derives a recursive algorithm for computing an empirical measure of independence. Its recursive nature, combined with the proposed data sparsification front-end, means its computational complexity and memory requirements are significantly lower than those of its competitors. This allows it to be applied to large datasets such as those occurring in neuroscience, where one seeks to understand which areas of the brain communicate with which other areas.

Earlier Works. The underlying challenge addressed by this paper is how to reduce the computational complexity of working with high-dimensional Gram matrices. To date, popular approaches include the Nyström method and low-rank approximations (as provided by the incomplete Cholesky decomposition [4]). Many of the usual techniques are detailed in classical monographs such as [13, 11]. For the particular application of computing the HSIC (Section 3), the computational complexity was reduced by applying the incomplete Cholesky decomposition [7, 14]. We have taken a different approach

P. O. Amblard is funded by a Marie Curie International Outgoing Fellowship from the European Community

and derived a *recursive* version of the algorithm. It appears to have lower complexity than competing methods.

Dictionary-based data sparsification schemes have been proposed especially for the design of on-line algorithms [3, 8, 12]. Here, we give a vector quantisation perspective to dictionary-based data sparsification, and use this perspective to motivate an application to the estimation of mean elements [9]. The idea of quantisation has also been used very recently in the context of the kernel LMS algorithm [2]. It has also occurred in a very different context closely linked to information theory, as for example in [10].

2. SPARSIFICATION AND VECTOR QUANTISATION

Coherence based sparsification. Consider a problem involving a data set $\mathcal{S}_n = \{x_i, i = 1, \ldots, n\}$ of i.i.d. random vectors defined on some probability space, and taking values in an appropriate metric space, say $\mathbb{R}^{n_x}$. The problem can be an inference problem, such as estimating a parameter from the data set, or finding a functional link between some components of the vectors, or taking some decision based on the observed data. In any case, we assume that we deal with an approach where the data are embedded into a reproducing kernel Hilbert space (rkHs) $\mathcal{H}_x$ using the kernel $k_x : \mathbb{R}^{n_x} \longrightarrow \mathcal{H}_x$. For a review of kernel methods and rkHs we refer to [11, 13, 15]

The sparsification technique based on the coherence is the following. It consists in a recursive construction of a dictionary, according to which new data are added if and only if they are deemed to lack sufficient coherence with the elements of the dictionary. Let $\mathcal{D}_n$ be a subset of size $s(n)$ of $\{1, \ldots, n\}$. We index the elements of the dictionary by greek letters. The dictionary contains the time index of the variable retained. For $n = 1$ we simply set $\mathcal{D}_1 = \{1\}$. Let μ be a real number less than or equal to 1. The dictionary is recursively grown by the simple rule

$$\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{n\} \Longleftrightarrow \max_{\alpha \in \mathcal{D}_{n-1}} \frac{\big|k_x(x, x_\alpha)\big|}{\sqrt{k_x(x,x)k_x(x_\alpha, x_\alpha)}} < \mu \qquad (1)$$

The kernel is in the sequel chosen to be of unit norm. This simplifies the criterion since for any x, this means $k_x(x, x) = 1$. This criterion proposed by Richard in [12] is a simplification of the approximate linear dependence (ALD) criterion of [3]. The latter is based on linear estimation of a candidate $k_x(., x_n)$ from the members $\{k_x(., x_\alpha), \alpha \in \mathcal{D}_{n-1}\}$ of the dictionary. Since the fundamental ingredient of linear estimation is correlation, a candidate $k_x(., x_n)$ is likely to be added to the dictionary by ALD if the correlations $\big\langle k_x(., x_n) \big| k_x(., x_\alpha) \big\rangle, \alpha \in \mathcal{D}_{n-1}$ are small. But thanks to the reproducing property, these correlations are nothing but $k_x(x_\alpha, x_n), \alpha \in \mathcal{D}_{n-1}$, whence the coherence criterion. Note that this criterion have close connection with the criterion used in [8] for the Kernel LMS. As shown in [12], the size of the dictionary is ensured to remain finite as long as the data live in some compact subspace. Therefore, theoretically this precludes the application of the technique to unbounded data (Gaussian random variables say). Practically however, the growth rate of the dictionary for unbounded data appears very slow since it is dominated at long times by low probabilities events. In the sequel, we restrict the discussion to radial kernels $k(x, y) = \varphi\big(d(x, y)\big)$, φ being a strictly decreasing function from $\mathbb{R}^+$ to itself with $\varphi(0) = 1$; d is a metric on the input space (typically the Euclidean distance in $\mathbb{R}^k$ for signal processing applications). This restriction includes the widely used Gaussian and exponential kernels.

Vector quantisation perspective. The restriction of the kernels considered to radial kernels allows a simplification of the coherence criterion, namely

$$\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{n\} \Longleftrightarrow \min_{\alpha \in \mathcal{D}_{n-1}} d(x, x_\alpha) > \varphi^{-1}(\mu) \qquad (2)$$

This means that a candidate $k_x(., x_n)$ is added to the dictionary if and only if the datum x is sufficiently far away from all the data whose indices are in the dictionary. Equivalently, this means that the datum x is not included into the dictionary if there exists an index $\alpha \in \mathcal{D}_{n-1}$ such that $x \in \mathcal{V}_\alpha := \{w : d(w, x_\alpha) \leq \varphi^{-1}(\mu)\}$. As recalled earlier, if the data live in a compact subset of the input space, the the size of the dictionary remains finite as n goes to infinity. This fact is easily understood with the equivalent formulation. When the final size s of the dictionary which depends on μ and φ is attained, the set $\mathcal{V}_\alpha$ is nothing but a Voronoï cell defined by the metric d and the set $\mathcal{D}_n$.

In an algorithm where a dictionary as built above is used, the data set is thus reduced to the subset of samples whose indices belong to the dictionary. Therefore, if a new datum is to be processed, it will be compared using k_x only to the $x_\alpha, \alpha \in \mathcal{D}_n$. In this spirit, the sparsification is a vector quantisation of the input space. Furthermore, this quantisation is adaptive, and in the limit, produces a grid which is very much like a regular grid. Indeed, space is filled with points which are all separated by a distance at least $\varphi^{-1}(\mu)$, but no more than $2\varphi^{-1}(\mu)$, otherwise there would be some space to add a new vector to the dictionary. This is illustrated in Figure (1). As seen in the figure, the quantisation provides an almost regular grid of points selected adaptively, and almost independently of the distribution of the data. By comparison, a usual optimal quantisation in the L^2 sense would provide a partition more closely adapted to the distribution of the data.

Thus, if used alone, the partition obtained after sparsification looses a lot of information from the distribution. Therefore, if coherence-based sparsification is used, we can imagine obtaining some gain in the processing by including in the algorithms the empirical distribution based on the partition. This is done by defining $\pi_n(\alpha) = \sum_{i=1}^n \mathbf{1}(x_i \in \mathcal{V}_\alpha)$ and using $n^{-1}\pi_n(\alpha)$ as the empirical measure of the data based on the adaptive partition.

Adapting the dictionary, a simple example. To illustrate this, consider the problem of estimating the mean element of a rkHs $\mathcal{H}_x$, defined as the function $m_x(.) = E[k_x(., x)]$, where E stands for the expectation over the distribution of x. For information about random variables having values in Hilbert spaces, we refer to [5], and to [1] for the particular case of rkHS. To estimate the mean element, we can use the empirical estimator $me_x^n(.) = n^{-1}\sum_i k(., x_i)$. Practically, evaluating $me_x^n(.)$ at a point u requires the computation of $k(u, x_i)$ for all i included in the sum. In some applications this could be problematic when the number of data is large and must be stored for later use. However, if a sparsification procedure has been applied, storage requirements may become manageable. This motivates the following. We suppose having already the partition $\mathcal{D}$ of the data based on the coherence criterion. This amounts to forgetting an initial period of time during which the learning of the dictionary is done. Then we compare two estimators $m_{i,x}^n = \sum_{\alpha \in \mathcal{D}_n} k(., x_{i,\alpha})\frac{\pi_n(\alpha)}{n}$, $i = 1, 2$, where $x_{1,\alpha} = x_\alpha$ and $x_{2,\alpha} = \bar{x}_\alpha = \pi_n(\alpha)^{-1}\sum_{k \geq 1} x_k \mathbf{1}(x_k \in \mathcal{V}_\alpha)$. In the first estimator, all elements falling into the cell $\mathcal{V}_\alpha$ are replaced by the center of the cell x_α. In the second one, we use the knowledge of vector quantisation to replace the center by the estimated centroïd (or k-means) of the cell. For any $f \in \mathcal{H}_x$, we evaluate $\big\langle f \big| m_{i,x} \big\rangle$ to obtain an estimation of $E[f(x)]$. Then it can be shown that the conditional bias of both estimators is mainly due to quantisation errors in the quadrature $E[f(x)] \approx \sum_{\alpha \in \mathcal{D}_n} k(., x_c) P_x(\mathcal{V}_\alpha)$ where

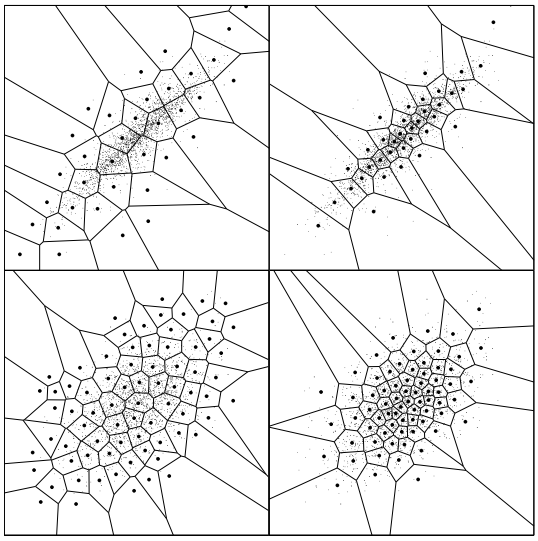

Fig. 1. Illustration of the vector quantisation induced by the coherence criterion, and comparison to usual vector quantisation (big dots are centers of the cells, small dots are the data points). Voronoï cells associated to the coherence sparsification (**Left**) and to the optimal (L^2) vector quantisation (**Right**). **Up:** Gaussian in x, Exponential in y data. **Down:** Gaussian data

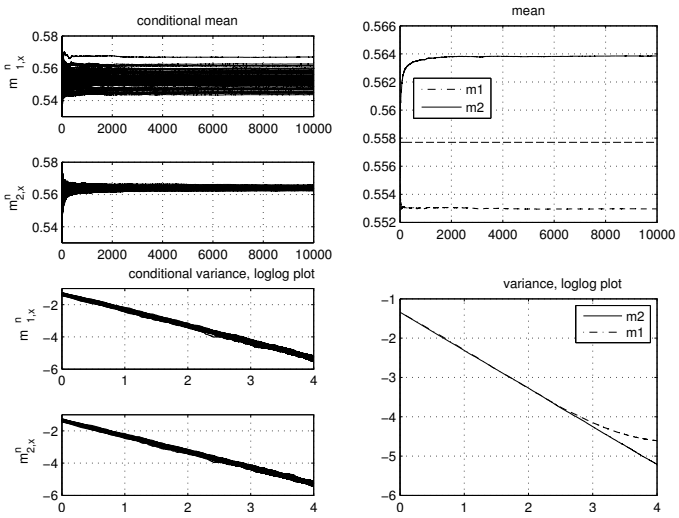

Fig. 2. Illustrations of the statistics of the sparse estimate of the mean. Two upper left plots: 100 realizations of the conditional mean illustrating the one order of magnitude difference between the two estimators. Upper right plot: the mean of the estimates are of the same order. Lower left plots: the conditional variances are of the same order. Lower right plot: for high n, the variance of the estimator based on the k means is superior to the other. Note that lower plots are log-log plots.

$x_c = x_\alpha$ is the center of the cell for estimator 1, and $x_c = \bar{x}_\alpha$ is the centroïd of the cell for estimator 2. It is well-known from numerical analysis that taking the center to be the centroïds gives a gain of one order of magnitude in the convergence rate of this approximation. This gain occurs here conditionally to the partition, and the fluctuations of the conditional mean will be greater for estimator 1. However, their bias are the same. We can show that the conditional variances are of the same order of magnitude. Therefore, thanks to $\mathrm{Var}[m] = E[\mathrm{Var}[m|\mathcal{D}]] + \mathrm{Var}[E[m|\mathcal{D}]]$, the variance of estimator 2 is lower than that of estimator 1. The proofs are mainly done as in the evaluation of errors in quadrature methods. The functions to be integrated are assumed to have some degrees of smoothness, and expansions are considered. All the calculations are done conditional on the partition.

To illustrate, we apply the previous estimators to $E[\exp(-x^2)]$ when X is uniformly distributed over $[0,1]^{n_x}$. Then the mean is $(\sqrt{\pi}\,\mathrm{erf}(1)/2)^k$ where erf stands for the usual error function. We generated $N_p = 100$ random partitions $\mathcal{D}_k$, $k = 1, \ldots, N_p$, for a given $\mu = 0.9$ calculated from series of length 10000, so that we can assume that the support is completely covered, or equivalently that the dictionary size has reached its limit. Then for each partition we generate $N_r = 100$ snaphshots of $n = 10000$ samples for which we evaluate the two estimators. Averaging over the N_r snapshots allows us to have N_p realizations of the estimate of the conditional mean and of the conditional variance as a function of n. Precisely, we calculate $m^n_{i,x,k,l}$ and $m^n_{2,x,k,l}$ for $k = 1, \ldots, N_p$ and $l = 1, \ldots, N_r$. Then $N_r^{-1} \sum_l m n^n_{1,x,k,l}$ (resp. $m^n_{2,x,k,l}$) is an estimate of $E[m^n_{1,x}|\mathcal{D}_k]$ (resp. $E[m^n_{2,x}|\mathcal{D}_k]$). Likewise, $N_r^{-1} \sum_l (m^n_{1,x,k,l} - N_r^{-1} \sum_l m^n_{1,x,k,l})^2$ is an estimate of $\mathrm{Var}[m^n_{1,x}|\mathcal{D}_k]$ (the same holds true for $m^n_{1,x}$ as well). We plot some results in Figure (2). The advantage of using the centroïds instead of the center is illustrated by the upper left plots and the lower right plot. In the upper left plots, we clearly see that the conditional mean has a worse dispersion for mn^n_x, which is the meaning of the order of magnitude difference between the two estimator. This implies that the variance of $m^n_{1,x}$ is bigger than the variance of $m^n_{2,x}$, even if the conditional variances are of the same order, as mentioned above.

3. APPLICATION TO HSIC

We now illustrate the application of the sparsification technique to the estimation of a measure of independence proposed by Gretton [6]. The measure is named HSIC for Hilbert-Schmidt Independence Criterion. As developed in [5], it is possible to define the covariance and cross-covariance of random variables taking values in Hilbert spaces. Precisely, if we embed two random vectors x and y defined on a common probability space into two different rkHs $\mathcal{H}_x$ and $\mathcal{H}_y$ using respectively two kernels $k_u : \mathbb{R}^{n_u} \times \mathbb{R}^{n_u} \to \mathbb{R}, u = x, y$, then the cross-covariance operator is defined as the unique linear bounded operator $\Sigma_{yx} : \mathcal{H}_x \longrightarrow \mathcal{H}_y$ such that $\langle g|\Sigma_{yx} f\rangle_{\mathcal{H}_y} = \mathrm{Cov}\,[f(x), g(y)]$. This operator is well-defined provided $E[k_x(x,x)] < +\infty$ and $E[k_y(y,y)] < +\infty$. Then the operator is known to be Hilbert-Schmidt, meaning that its Hilbert-Schmidt norm defined as $\|\Sigma_{yx}\|^2 := \sum_i \left\|\Sigma_{yx}\varphi_i\right\|^2_{\mathcal{H}_y}$ is finite, for any orthonormal basis $\{\varphi_i\}_{i\in\mathbb{N}}$.

A theorem proved by Gretton extends Rényi's theorem stating that independence is equivalent to $\mathrm{Cov}\,[f(x), g(y)] = 0$ for all continuous bounded functions f and g. Gretton's theorem states that if the kernel is universal (in the sense that the RKHS associated to it is dense in the set of continuous bounded functions), then $\sup_{f\in\mathcal{U}_x, g\in\mathcal{U}_y} \langle g|\Sigma_{yx} f\rangle_{\mathcal{H}_y} = 0$ is equivalent to independence between x and y. Here, $\mathcal{U}$ stands for the unit ball of $\mathcal{H}$, the subset of functions with norm less than 1. The magic with this result is that the quantity involved in this result is nothing but the usual norm of the covariance operator which can be efficiently estimated when dealing with data. Furthermore, this norm is known to be less than or equal to

the Hilbert-Schmidt norm. Therefore Gretton's result remains valid when dealing with the Hilbert-Schmidt norm, a quantity even easier to estimate from data [7, 6]. Finally, there are well-known universal kernels, such as the Gaussian and the exponential kernels.The calculation of the estimate can be done in $O(n^2)$ steps for data length n, a high complexity which can be lowered to a linear complexity in n by using low-rank approximations of Gram matrices. However, the complexity depends on the rank obtained which is often not very small compared to n.

Here we adopt another strategy by giving a recursive implementation of HSIC, and then by applying to the recursive implementation the coherence based sparsification technique. In the full recursive implementation, the Gram and centering matrices are not explicitly used. This lowers memory requirements and significantly reduces the required number of floating-point operations. The complexity remains however $O(n^2)$, but the gain in storage and number of operations makes the implementation much more rapid than an implementation using matrices. Furthermore, applying jointly the sparsification procedure allows to gain much more.

Given data x_i, y_i, for $i = 1, \ldots, n$, empirical estimators of $\|\Sigma_{YX}\|^2$ can be obtained using $n \times n$ Gram matrices K_x^n and K_y^n, whose (i,j)th entries are respectively $k_x(x_i, x_j)$ and $k_y(y_i, y_j)$. Then the HSIC estimate is given by $n^{-2}\mathrm{Tr}\left(K_x^n C_n K_y^n C_n\right)$ with $C_n := I_n - \mathbf{1}\mathbf{1}^\top/n$, $\mathbf{1}$ is a vector of 1 of size n. Note that this estimator is only asymptotically unbiased. An unbiased estimate can be designed (see [14]), and the following development can be done for the unbiased case but is not presented for the sake of simplicity. Although it is possible to design a recursive algorithm based on the previous form, the following alternative derivation has the advantage of being able to work well with the sparsification technique introduced in Section 2.

The required empirical estimates are the empirical mean elements in the rkHs and the cross-moment operator. We present them jointly with the sparsification procedure. Suppose at time n the dictionary is $\mathcal{D}_n$, and the number of samples already in cell $\mathcal{V}_\alpha$ given by $\pi_n(\alpha)$. Then the estimators of the mean elements and the cross-moment are respectively $m_x^n = n^{-1}\sum_{\alpha\in\mathcal{D}_n}\pi_n(\alpha)k(.,x_\alpha)$ and $M_{yx}^n f = n^{-1}\sum_{\alpha\in\mathcal{D}_n}\pi_n(\alpha)\langle f|k(.,x_\alpha)\rangle k(.,y_\alpha)$. This allows the cross-covariance to be estimated as $C_{yx}^n f = M_{yx}^n f - \langle f|m_x^n\rangle m_y^n$. Let $s(n)$ be the size of the dictionary at time n. Define $\boldsymbol{\pi}_{n-1}$ to be the vector with entries $\pi_{n-1}(\alpha), \alpha \in \mathcal{D}_{n-1}$. Let $\boldsymbol{k}_x^n$ be the vector with entries $k_x(x_n, x_\alpha), \alpha \in \mathcal{D}_{n-1}$. Let $(A \circ B)_{ij} := A_{ij}B_{ij}$ In the sequel, $z = (x, y)$ and $k_z = k_x \otimes k_y$ is the tensor product of the kernels, and is the kernel of the tensor product between $\mathcal{H}_x$ and $\mathcal{H}_y$ [15]. The recursive sparse form of HSIC is given by

$$
\begin{aligned}
\|C_{yx}^n\|^2 &= \|M_{yx}^n\|^2 + \|m_x^n\|^2\|m_y^n\|^2 - 2c_{yx}^n \\
\|M_{yx}^n\|^2 &= \frac{(n-1)^2}{n^2}\|M_{yx}^{n-1}\|^2 + \frac{2}{n^2}\boldsymbol{\pi}_{n-1}^\top \boldsymbol{k}_x^n \circ \boldsymbol{k}_y^n + \frac{\|k_x\|^2\|k_y\|^2}{n^2} \\
\|m_x^n\|^2 &= \frac{(n-1)^2}{n^2}\|m_x^{n-1}\|^2 + \frac{2}{n^2}\boldsymbol{\pi}_{n-1}^\top \boldsymbol{k}_x^n + \frac{\|k_x\|^2}{n^2} \\
c_{yx}^n &= \frac{1}{n^3}\boldsymbol{\pi}_n^\top \boldsymbol{v}_x^n \circ \boldsymbol{v}_y^n \text{ with :}
\end{aligned}
$$

1. if $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{n\} \iff \max_{\alpha\in\mathcal{D}_{n-1}}\left|k_z(z_n, z_\alpha)\right| < \mu$:

$$
\boldsymbol{v}_x^n = \begin{pmatrix} \boldsymbol{v}_x^{n-1} + \boldsymbol{k}_x^n \\ \boldsymbol{\pi}_{n-1}^\top \boldsymbol{k}_x^n + \|k_x\|^2 \end{pmatrix} \text{ and } \boldsymbol{\pi}_n = \begin{pmatrix} \boldsymbol{\pi}_{n-1} \\ 1 \end{pmatrix}
$$

2. if $\mathcal{D}_n = \mathcal{D}_{n-1}$: $a = \arg\max_{\alpha\in\mathcal{D}_{n-1}}\left|k_{xy}(z_\alpha, z_n)\right|$,

$$
\boldsymbol{v}_x^n = \boldsymbol{v}_x^{n-1} + \boldsymbol{k}_x^n \text{ and } \boldsymbol{\pi}_n = \boldsymbol{\pi}_{n-1} + \delta_{a\alpha}
$$

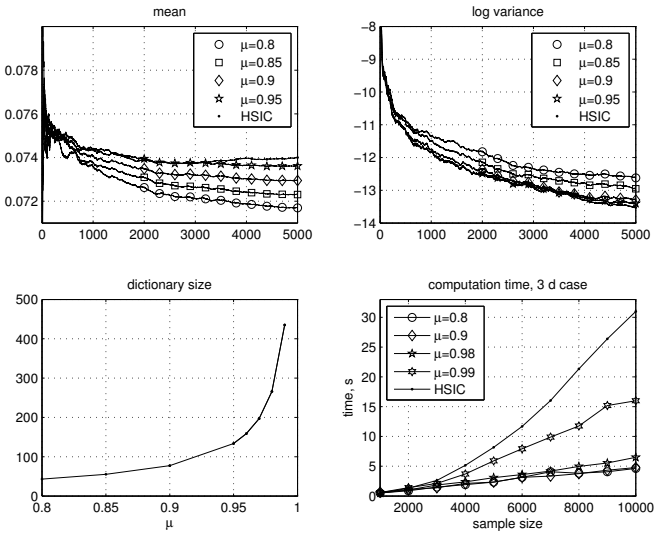

Fig. 3. Sparse HSIC to assess independence between a 1d Gaussian variable and a 1d Exponential random variable. **Up Left** and **right:** Mean and log variance of the recursive algorithm function of sample size n. **Bottom Left:** Mean size of the dictionary as a function of μ. **Bottom Right:** Computation time as a function of the sample size for assessing independence between 3d Gaussian vectors.

and likewise for the estimates indexed by y. In these expressions, $\|k_x\|^2$ stands for $k_x(x_n, x_n)$ and likewise for y. In item 1 above, if the coherence test is passed the dictionary is updated. Therefore its dimension is increased by one unit, and the vector $\boldsymbol{\pi}_n$ updated accordingly. When the coherence test fails, the dictionary remains as it is at time $n-1$, the dimension of $\boldsymbol{\pi}$ is not increased, only the components a corresponding to the index of the cell where x_n falls is increased by 1. Note that the coherence test is performed in the tensor product space $\mathcal{H}_x \otimes \mathcal{H}_y$.

Let us illustrate the behavior of the algorithm on a simulation. We study the dependence between the components of a 2d rotated vector. The first component of the initial vector is a standard Gaussian variable, the second component is a bilateral Exponential variable with parameter 1. We evaluate HSIC and its sparse version for $\mu = 0.8, 0.85, 0.9, 0.95$. The kernel used is the Gaussian $\exp(-\|z\|^2/1.2)$. We calculate the result for 100 realizations of 5000 samples each.The mean convergence and the variance over time are plotted in Figure (3), top row. As μ approaches 1, the sparse version approaches the full HSIC algorithm. Further, the error made compared to HSIC is low, and the loss in variance is very low. This is remarkable since for example $\mu = 0.95$, the dictionary has a size of only 140, compared to 5000 samples, which which is a significant reduction in computational effort (see the left plot in the bottom row in figure (3)). To illustrate the gain obtained in computation time, we have tested independence between two correlated 3 dimensional random vectors. The recorded time of computation as evaluated in our implementation is plotted in the bottom right plot in figure (3) as a function of the sample size for several values of μ. A seen, for small values of μ, the behavior is almost linear in the sample size.

4. REFERENCES

[1] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers, 2004.

[2] B. Chen, S. Zhao, P. Zhu, and J. Principe. Quantized kernel least mean square algorithm. *IEEE Trans. on Neural Networks and Learning Systems*, 23(1):22–32, 2012.

[3] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Trans. on Signal Processing*, 52(8):2275–2284, 2004.

[4] S. Fine and K. Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

[5] R. Fortet. *Vecteurs, fonctions et distributions aléatoires dans les espaces de Hilbert (Random Vectors, functions and distributions in Hilbert spaces)*. (in French), Hermès, 1995.

[6] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. In Springer-Verlag, editor, *Proceedings of ICALT*, pages 63–77, 2005.

[7] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

[8] W. Liu, P. Pokharel, and J. Principe. The kernel least mean squares algorithm. *IEEE Trans. on Signal Processing*, 56(2):543–554, 2008.

[9] G. Pagès. A space quantization method for numerical integration. *Jour. of Comp. Appl. Math.*, 89:1–38, 1997.

[10] A. V. Rao, D. J. Miller, K. Rose, and A. Gersho. A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Trans. on Patt. Anal. and Mach. Intel.*, 21(2):159–173, 1999.

[11] C. E. Rassmussen and C. K .I. Williams, *Gaussian processes for machine learning*, MIT Press, Cambridge, Ma, USA, 2006.

[12] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, Mar 2009.

[13] B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT Press, Cambridge, Ma, USA, 2002.

[14] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

[15] I. Steinwart and A. Christmann. *Support vector machines*. Springer, 2008.

[16] Y. Sun, F. Gomez, and J. Schmidhuber. On the size of the online kernel sparsification dictionary. In *proceedings of ICML, Edinburgh, Scotland*, 2012.