

# Kernelized Sorting

Novi Quadrianto, Alex J. Smola, Le Song, and Tinne Tuytelaars

**Abstract**—Object matching is a fundamental operation in data analysis. It typically requires the definition of a similarity measure between the classes of objects to be matched. Instead, we develop an approach which is able to perform matching by requiring a similarity measure only *within* each of the classes. This is achieved by maximizing the dependency between matched pairs of observations by means of the Hilbert-Schmidt Independence Criterion. This problem can be cast as one of maximizing a quadratic assignment problem with special structure and we present a simple algorithm for finding a locally optimal solution.

**Index Terms**—Sorting, matching, kernels, object alignment, Hilbert-Schmidt Independence Criterion.

## 1 INTRODUCTION

MATCHING pairs of objects is a fundamental operation of unsupervised learning. For instance, we might want to match a photo with a textual description of a person, a map with a satellite image, or a music score with a music performance. In these cases, it is desirable to have a compatibility function which determines how one set may be translated into the other. For many such instances, we may be able to *design* a compatibility score based on prior knowledge or to observe one based on the co-occurrence of such objects. This has led to good progress in areas such as graph matching [2], [3], [4].

In some cases, however, such a match may not exist or it may not be given to us beforehand. That is, although we may have a good understanding of two sources of observations, say  $\mathcal{X}$  and  $\mathcal{Y}$ , we may not understand the mapping between the two spaces. For instance, we might have two collections of documents purportedly covering the same content, written in two different languages. Here, it should be our goal to determine the correspondence between both sets and identify a mapping between the two domains [5]. In yet other cases, matching by minimization of a distance function is a popular strategy for point assignment [6], [7], [8].

In the following, we present a method which is able to perform such matching *without* the need of a cross-domain similarity measure, and we shall show that if such measures exist, it generalizes existing approaches. Our method relies on

the fact that one may estimate the *dependence* between sets of random variables even without knowing the cross-domain mapping. Various dependence criteria are available. We choose the Hilbert-Schmidt Independence Criterion (HSIC) between two sets and maximize over the permutation group to find a good match. As a side effect, we obtain an explicit representation of the covariance.

We show that our method generalizes sorting. When using a different measure of dependence, namely, an approximation of the mutual information, our method is related to an algorithm proposed by Jebara [5]. Finally, we give a simple approximation algorithm for Kernelized Sorting, and we discuss how a number of existing algorithms fall out as special cases of the Kernelized Sorting method.

### 1.1 Sorting and Matching

The basic idea underlying our algorithm is simple. Denote by  $X = \{x_1, \dots, x_m\} \subseteq \mathcal{X}$  and  $Y = \{y_1, \dots, y_m\} \subseteq \mathcal{Y}$  two sets of observations between which we would like to find a correspondence. That is, we would like to find some element  $\pi$  of the permutation group  $\Pi_m$  on  $m$  elements

$$\Pi_m := \{\pi | \pi \in \{0, 1\}^{m \times m} \text{ where } \pi 1_m = 1_m, \pi^\top 1_m = 1_m\},$$

such that the set of pairs  $Z(\pi) := \{(x_i, y_{\pi(i)}) \text{ for } 1 \leq i \leq m\}$  corresponds to maximally dependent random variables.<sup>1</sup> Here,  $1_m \in \mathbb{R}^m$  is the vector of all ones. We seek a permutation  $\pi$  such that the mapping  $x_i \rightarrow y_{\pi(i)}$  and its converse mapping from  $y$  to  $x$  are simple.

For a given measure  $D(Z(\pi))$  of the dependence between  $x$  and  $y$ , we define nonparametric sorting of  $X$  and  $Y$  as follows:

$$\pi^* := \underset{\pi \in \Pi_m}{\operatorname{argmax}} D(Z(\pi)). \quad (1)$$

This paper is concerned with measures of  $D$  and approximate algorithms for (1). In particular, we will investigate the Hilbert-Schmidt Independence Criterion, as well as the mutual information.

The remainder of the paper is organized as follows: We first explain in detail the Hilbert-Schmidt Independence Criterion and how it can be used for Kernelized Sorting (Section 2). Next, we discuss the problem of optimization

1. We use  $\pi(i)$  to denote permutation mapping of the  $i$ th element and  $\pi$  to denote permutation matrix whose entries are all 0, except that in row  $i$ , the entry  $\pi(i)$  equals 1.

- N. Quadrianto is with the RSISE, Australian National University and the Statistical Machine Learning Group, Canberra Research Laboratory, NICTA, Tower A, 7 London Circuit, Canberra City, ACT 2601, Australia. E-mail: novi.quad@gmail.com.
- A.J. Smola is with Yahoo! Research, 4401 Great America Parkway, Santa Clara, CA 95054. E-mail: alex@smola.org.
- L. Song is with the Lane Center for Computational Biology, GHC7416, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213. E-mail: lesong@cs.cmu.edu.
- T. Tuytelaars is with K.U. Leuven ESAT-PSI, Kasteelpark Arenberg 10, bus 2441, B-3001 Leuven, Belgium. E-mail: tinne.tuytelaars@esat.kuleuven.be.

Manuscript received 5 Oct. 2008; revised 24 June 2009; accepted 20 Aug. 2009; published online 18 Nov. 2009.

Recommended for acceptance by D. Schuurmans.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-10-0667.

Digital Object Identifier no. 10.1109/TPAMI.2009.184.

(Section 3). In Section 4, a multivariate extension is proposed. Section 5 describes links with related work, and Section 6 shows possible applications, including data visualization, matching, and estimation. Section 7 concludes the paper.

## 2 HILBERT SPACE METHODS

Let sets of observations  $X$  and  $Y$  be drawn jointly from some probability distribution  $\Pr_{xy}$ . The HSIC [9] measures the dependence between  $x$  and  $y$  by computing the norm of the cross-covariance operator over the domain  $\mathcal{X} \times \mathcal{Y}$  in Hilbert Space. It can be shown, provided the Hilbert Space is characteristic [10], that this norm vanishes if and only if  $x$  and  $y$  are independent. A large value suggests strong dependence with respect to the choice of kernels.

### 2.1 Hilbert-Schmidt Independence Criterion

Formally, let  $\mathcal{F}$  be the Reproducing Kernel Hilbert Space (RKHS) [11] on  $\mathcal{X}$  with associated kernel  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  and feature map  $\phi : \mathcal{X} \rightarrow \mathcal{F}$ . Let  $\mathcal{G}$  be the RKHS on  $\mathcal{Y}$  with kernel  $l$  and feature map  $\psi$ . The cross-covariance operator  $\mathcal{C}_{xy} : \mathcal{G} \mapsto \mathcal{F}$  is defined by [12] as

$$\mathcal{C}_{xy} = \mathbf{E}_{xy}[(\phi(x) - \mu_x) \otimes (\psi(y) - \mu_y)], \quad (2)$$

where  $\mu_x = \mathbf{E}[\phi(x)]$ ,  $\mu_y = \mathbf{E}[\psi(y)]$ , and  $\otimes$  is the tensor product. HSIC, denoted as  $\mathcal{D}$ , is then defined as the square of the Hilbert-Schmidt norm of  $\mathcal{C}_{xy}$  [9] via  $\mathcal{D}(\mathcal{F}, \mathcal{G}, \Pr_{xy}) := \|\mathcal{C}_{xy}\|_{\text{HS}}^2$ . In term of kernels, HSIC can be expressed as

$$\begin{aligned} \|\mathcal{C}_{xy}\|_{\text{HS}}^2 &= \mathbf{E}_{xx'yy'}[k(x, x')l(y, y')] \\ &\quad + \mathbf{E}_{xx'}[k(x, x')]\mathbf{E}_{yy'}[l(y, y')] \\ &\quad - 2\mathbf{E}_{xy}[\mathbf{E}_{x'}[k(x, x')]\mathbf{E}_{y'}[l(y, y')]], \end{aligned} \quad (3)$$

where  $\mathbf{E}_{xx'yy'}$  is the expectation over both  $(x, y) \sim \Pr_{xy}$  and an additional pair of variables  $(x', y') \sim \Pr_{xy}$  drawn independently according to the same law. Given a sample  $Z = \{(x_1, y_1), \dots, (x_m, y_m)\}$  of size  $m$  drawn from  $\Pr_{xy}$ , an empirical estimate of HSIC is given by

$$D(\mathcal{F}, \mathcal{G}, Z) = (m-1)^{-2} \text{tr } HKHL = (m-1)^{-2} \text{tr } \bar{K} \bar{L}, \quad (4)$$

where  $K, L \in \mathbb{R}^{m \times m}$  are the kernel matrices for the set  $X$  and the set  $Y$ , respectively, i.e.,  $K_{ij} = k(x_i, x_j)$  and  $L_{ij} = l(y_i, y_j)$ . Moreover,  $H_{ij} = \delta_{ij} - m^{-1}$  centers the observations of set  $X$  and set  $Y$  in feature space. Finally,  $\bar{K} := HKH$  and  $\bar{L} := HLH$  denote the centered versions of  $K$  and  $L$ , respectively. Note that (4) is a *biased* estimate where the expectations with respect to  $x, x', y$ , and  $y'$  have all been replaced by empirical averages over the set of observations (for further properties of this empirical estimator, refer to [9, Theorem 7] and references therein). This is acceptable in most situations. For an unbiased estimate which addresses problems in situations where the main diagonal terms in  $K$  and  $L$  dominate, see Section 2.3.

### 2.2 Kernelized Sorting

Previous work used HSIC to *measure* independence between given random variables [9]. Here, we use it to *construct* a mapping between  $X$  and  $Y$  by permuting  $Y$  to maximize dependence. There are several advantages in using HSIC as

a dependence criterion. First, HSIC satisfies concentration of measure conditions [9]. That is, for random draws of observation from  $\Pr_{xy}$ , HSIC provides values which are very similar. This is desirable, as we want our mapping to be robust to small changes. Second, HSIC is easy to compute since only the kernel matrices are required and no density estimation is needed. The freedom of choosing a kernel allows us to incorporate prior knowledge into the dependence estimation process. The consequence is that we are able to generate a family of methods by simply choosing appropriate kernels for  $X$  and  $Y$ .

**Lemma 1.** *With  $D(Z(\pi)) = D(\mathcal{F}, \mathcal{G}, Z)$  as in (4), the nonparametric sorting problem is given by*

$$\pi^* = \underset{\pi \in \Pi_m}{\operatorname{argmax}} \operatorname{tr} \bar{K} \pi^\top \bar{L} \pi. \quad (5)$$

**Proof.** We only need to establish that  $H\pi^\top = \pi^\top H$  since the rest follows immediately from the definition of (4). Since  $\pi 1_m = 1_m$  and  $\pi^\top 1_m = 1_m$ , then

$$\begin{aligned} H\pi &= \left( I_m - \frac{1}{m} 1_m 1_m^\top \right) \pi = \left( \pi - \frac{1}{m} 1_m 1_m^\top \pi \right) = \left( \pi - \frac{1}{m} 1_m 1_m^\top \right) \\ &= \left( \pi - \frac{1}{m} \pi 1_m 1_m^\top \right) = \pi \left( I_m - \frac{1}{m} 1_m 1_m^\top \right) = \pi H. \end{aligned}$$

Hence,  $H$  and  $\pi$  matrices commute.  $\square$

Note that the optimization problem (5) is in the form of Koopmans-Beckmann equation [13] and is, in general, NP-hard, as it is an instance of a quadratic assignment problem [14]. Nonetheless, the objective function is indeed reasonable. We demonstrate this by proving that sorting is a special case of the optimization problem set out in (5). For this, we need the following inequality due to Polya, Littlewood, Hardy, and Blackwell [15]:

**Lemma 2.** *Let  $a, b \in \mathbb{R}^m$ , where  $a$  is sorted ascendingly. If  $\operatorname{argsort} b$  denotes the vector of ranks of ascendingly sorted entries of vector  $b$ , then  $a^\top \pi b$  is maximized for  $\pi = \operatorname{argsort} b$ .*

Consider the case of scalar random variables and a linear kernel.

**Lemma 3.** *Let  $\mathcal{X} = \mathcal{Y} = \mathbb{R}$  and let  $k(x, x') = xx'$  and  $l(y, y') = yy'$ . Moreover, assume that  $x$  is sorted ascendingly. In this case, (4) is maximized by either  $\pi = \operatorname{argsort} y$  or by  $\pi = \operatorname{argsort} -y$ .*

**Proof.** Under the assumptions, we have that  $\bar{K} = Hxx^\top H$  and  $\bar{L} = Hyy^\top H$ . Hence, we may rewrite the objective as  $[(Hx)^\top \pi (Hy)]^2$ . This is maximized by sorting  $Hy$  ascendingly. Since the centering matrix  $H$  only changes the offset, but not the order, this is equivalent to sorting  $y$ . We have two alternatives since the objective function is insensitive to sign reversal of  $y$ .  $\square$

This means that sorting is a special case of Kernelized Sorting; hence the name. The ambiguity in the solution of the optimization problem arises from the fact that instead of having direct access to the entries  $x_i$ , we only access them by means of the kernel  $k(x_i, x_j)$ . In this context, changes of all observations via  $x \leftarrow -x$  leave the kernel unchanged; hence they cannot be detected in the sorting operation.

When solving the general problem, it turns out that a projection onto the principal eigenvectors of  $\bar{K}$  and  $\bar{L}$  is a good initialization of an optimization procedure.

### 2.3 Diagonal Dominance

In some cases, the biased estimate of HSIC, as given by (4), leads to rather undesirable results, in particular in the case of document analysis. This arises from the fact that kernel matrices on texts tend to be diagonally dominant: A document tends to be *much* more similar to itself than to others; hence the values of the diagonal entries  $K_{ii}$  considerably exceed those of the off-diagonal terms. In this case, the  $O(1/m)$  bias of (4) is significant. After all, it is due to the terms in  $\text{tr}HKHL$  which contain matching index pairs  $\{ii\}$  with respect to  $K$  and  $L$  that are responsible for the bias. While their number is only  $O(m)$  (the total number of terms is  $O(m^2)$ ), they can still cause considerable damage on finite amounts of data.

Unfortunately, the minimum variance unbiased estimator [9] does not have a computationally appealing form. This can be addressed as follows at the expense of a slightly less efficient estimator with a considerably reduced bias: We replace the expectations (3) by sums where no pairwise summation indices are identical. This leads to the objective function

$$\begin{aligned} \frac{1}{m(m-1)} \sum_{i \neq j} K_{ij} L_{ij} + \frac{1}{m^2(m-1)^2} \sum_{i \neq j, u \neq v} K_{ij} L_{uv} \\ - \frac{2}{m(m-1)^2} \sum_{i,j \neq i,v \neq i} K_{ij} L_{iv}. \end{aligned}$$

This estimator still has a small degree of bias, albeit significantly reduced since it only arises from the product of expectations over (potentially) independent random variables. Using the shorthand  $\tilde{K}_{ij} = K_{ij}(1 - \delta_{ij})$  and  $\tilde{L}_{ij} = L_{ij}(1 - \delta_{ij})$  for kernel matrices where the main diagonal terms have been removed, we arrive at the expression  $(m-1)^{-2} \text{tr} \tilde{H} \tilde{K} \tilde{H} \tilde{L}$ . The advantage of this term is that it can be used as a drop-in replacement in Lemma 1 without any need for changing the optimization algorithm.

### 2.4 Stability Analysis

Before discussing practical issues of optimization, let us briefly study the statistical properties of the objective function. First, note that the solution  $\text{argmax}_\pi \text{tr} \bar{K} \pi^\top \bar{L} \pi$  is *not* stable under sampling in general. A simple example may illustrate this. Assume that  $X = \{1, 2, 3\}$  and  $Y = \{1, 2, 2 + \epsilon\}$ . In this case, the identity permutation  $[(1)(2)(3)]$  is sufficient for maximal alignment between  $X$  and  $Y$ . Now replace the third element in  $Y$ , that is,  $2 + \epsilon$  by  $2 - \epsilon$ . In this case, the permutation  $[(1)(2, 3)]$  which swaps the elements 2 and 3 is optimal. Nonetheless, by a suitable choice of  $\epsilon$ , we can make the change in the objective function arbitrarily small.

What we can prove, however, is that changes in the minimum value of the *objective* function are well controlled under the optimization procedure. This relies on McDiarmid's concentration inequality [16] and on the fact that the minima of close functions are close.

**Lemma 4.** Denote by  $f$  and  $g$  functions on a domain  $\mathcal{X}$  with  $|f(x) - g(x)| < \epsilon$  for all  $x \in \mathcal{X}$ . In this case,  $|\min_{x \in \mathcal{X}} f(x) - \min_{x \in \mathcal{X}} g(x)| < \epsilon$ .

**Proof.** Consider  $x^* = \arg \min_{x \in \mathcal{X}} g(x)$ , then  $|f(x^*) - g(x^*)| < \epsilon$ . Since  $f(x^*) \geq \min_{x \in \mathcal{X}} f(x)$ , then  $|\min_{x \in \mathcal{X}} f(x) - g(x^*)| \leq |f(x^*) - g(x^*)| < \epsilon$ .  $\square$

**Lemma 5 (Concentration Inequality [16]).** Denote by  $f : \mathcal{X}^m \rightarrow \mathbb{R}$  a function satisfying

$$|f(\dots, x_{i-1}, x, x_{i+1}, \dots) - f(\dots, x_{i-1}, x', x_{i+1}, \dots)| \leq c/m$$

for all  $x, x', x_i \in \mathcal{X}$ . Moreover, let  $\Pr$  be a distribution on  $\mathcal{X}$ . In this case, for  $X = \{x_1, \dots, x_m\}$  drawn from  $\mathcal{P}^m$ , we have that, with probability exceeding  $1 - 2 \exp(-m\epsilon^2/c^2)$ , the following bound holds:

$$|f(X) - \mathbf{E}_{X \sim \mathcal{P}^m}[f(X)]| \leq \epsilon. \quad (6)$$

**Lemma 6 (Stability of Optimal Alignment).** Denote by  $A(X, Y) := m^{-2} \text{argmin}_{\pi \in \Pi_m} \text{tr} \pi^\top \bar{K} \pi \bar{L}$  the minimum of the alignment objective function for the sets  $X$  and  $Y$ . Moreover, assume that the kernels  $k$  and  $l$  are bounded by  $|k(x, x')|, |l(y, y')| \leq R$ . In this case,  $|A(X, Y) - \mathbf{E}_{X, Y}[A(X, Y)]| \leq \epsilon$  holds with probability at least  $1 - 4 \exp(-m\epsilon^2/8R^2)$ .

**Proof.** The first step in the proof is to check that if we replace any  $x_i$  by some  $x'_i$  or, alternatively, some  $y_j$  by  $y'_j$ , the value of  $A(X, Y)$  only changes by  $2R/m$ . This can be seen by using the fact that HSIC can be seen as the difference between the joint and the marginal expectations of the feature map  $k(x, \cdot)l(y, \cdot)$ .

Second, to deal with the fact that we have expectations over  $X$  and  $Y$ , we apply the concentration inequality twice and chain the arguments. To guarantee a total deviation of at most  $\epsilon$ , we apply a bound of  $\epsilon/2$  to the deviation between the empirical average and the expectation  $\mathbf{E}_X$ , and one more between the expectations  $\mathbf{E}_X$  and  $\mathbf{E}_{X, Y}$ . Applying the union bound for the corresponding probabilities of failure proves the claim.  $\square$

The consequence of this analysis is that while the optimal assignment itself is not stable, at least the objective function has this desirable property, i.e., for random draws of observations from joint distribution, the objective function provides values which are very similar. This means that in practice also most assignments are rather stable when they come to subsampling. This is evident in the experiments of Section 6.2.3.

## 3 OPTIMIZATION

Quadratic assignment problems [13] are notoriously hard and have attracted a rather diverse set of algorithms from simulated annealing, tabu search, and genetic algorithms to ant colony optimization. Below we present a rather simple method which is guaranteed to obtain a locally optimal solution by exploiting convexity in the optimization problem. It is very simple to implement provided that a linear assignment solver is available.

### 3.1 DC Programming

To find a local maximum of the matching problem, we may take recourse to a well-known algorithm, namely, DC Programming [17], which, in machine learning, is also known as the Concave Convex Procedure [18]. It works as follows: For a given function

$$f(x) = g(x) - h(x),$$

where  $g$  is convex and  $-h$  is concave, a lower bound can be found by

$$f(x) \geq g(x_0) + \langle x - x_0, \partial_x g(x_0) \rangle - h(x). \quad (7)$$

This lower bound is concave and it can be maximized effectively over a convex domain. Subsequently, one finds a new location  $x_0$  and the entire procedure is repeated. For the problem in Lemma 1,  $h(x) = 0$ , and thus, the DC programming corresponds to a successive maximization of linear lower bounds.

**Lemma 7.** Define  $\pi$  as a doubly stochastic matrix (8). The function  $\text{tr}\bar{K}\pi^\top\bar{L}\pi$  is convex in  $\pi$ .

**Proof.** Since  $\bar{K}, \bar{L} \succeq 0$ , we may factorize them as  $\bar{K} = U^\top U$  and  $\bar{L} = V^\top V$ . Hence, by the circularity of the trace, we may rewrite the objective function as  $\|V\pi U^\top\|^2$  or as  $\|(U \otimes V)\text{vec}(\pi)\|^2$  with  $\text{vec}(\cdot)$  denoting stacking column vectors of a matrix. This is clearly a convex quadratic function in  $\pi$ .  $\square$

Note that the set of feasible permutations  $\pi$  is constrained in a unimodular fashion, that is, the set

$$P_m := \left\{ M \in \mathbb{R}^{m \times m}, \text{ where } M_{ij} \geq 0 \text{ and } \sum_i M_{ij} = 1 \text{ and } \sum_j M_{ij} = 1 \right\} \quad (8)$$

has only integral vertices, namely, admissible permutation matrices. This means that the following procedure will generate a succession of permutation matrices which will yield a local maximum for the assignment problem:

$$\pi_{i+1} \leftarrow (1 - \lambda)\pi_i + \lambda \underset{\pi \in P_m}{\operatorname{argmax}} [\text{tr}\bar{K}\pi^\top\bar{L}\pi]. \quad (9)$$

Here, choosing  $\lambda = 1$  in the last step will ensure integrality. The optimization subproblem is well known as a Linear Assignment Problem and effective solvers are freely available [19].

**Lemma 8.** The algorithm described in (9) for  $\lambda = 1$  terminates in a finite number of steps.

**Proof.** We know that the objective function may only increase for each step of (9). Moreover, the solution set of the linear assignment problem is finite. Hence, the algorithm does not cycle.  $\square$

#### 3.1.1 Nonconvex Objective Function

When using the bias-corrected version of the objective function, the problem is no longer guaranteed to be convex. In this case, we need to add a line-search procedure along  $\lambda \in [0, 1]$  which maximizes

$$\text{tr}H\tilde{K}H[(1 - \lambda)\pi_i + \lambda\hat{\pi}_i]^\top H\tilde{L}H[(1 - \lambda)\pi_i + \lambda\hat{\pi}_i], \quad (10)$$

with  $\hat{\pi}_i = \operatorname{argmax}_{\pi \in P_m} [\text{tr}\bar{K}\pi^\top\bar{L}\pi]$ . Since the function is quadratic in  $\lambda$ , we only need to check whether the search direction remains convex in  $\lambda$ ; otherwise, we may maximize the term by solving a simple linear equation.

#### 3.1.2 Initialization

Since quadratic assignment problems are, in general, NP-hard, we may obviously not hope to achieve an optimal solution. This means that a good initialization is critical for good estimation performance. This can be achieved by using Lemma 3. That is, if  $\bar{K}$  and  $\bar{L}$  only had rank 1, the problem could be solved by sorting  $X$  and  $Y$  in matching fashion. Instead, we use the projections onto the first principal vectors as initialization in our experiments.

### 3.2 Relaxation to a Constrained Eigenvalue Problem

Yet another alternative is to find an approximate solution of the problem in Lemma 1 by solving

$$\underset{\eta}{\operatorname{maximize}} \eta^\top M \eta \text{ subject to } A\eta = b. \quad (11)$$

Here, the matrix  $M = \bar{K} \otimes \bar{L} \in \mathbb{R}^{m^2 \times m^2}$  is given by the outer product of the constituting kernel matrices,  $\eta \in \mathbb{R}^{m^2}$  is a vectorized version of the permutation matrix  $\pi$ , and the constraints imposed by  $A$  and  $b$  amount to the polytope constraints imposed by  $\Pi_m$ . This approach has been proposed by Cour et al. [4] in the context of balanced graph matching.

Note that the optimization algorithm for (11) as proposed by Cour et al. [4] is suboptimal. Instead, it is preferable to use the exact procedure described in [20], which is also computationally somewhat more efficient. Nonetheless, the problem with the relaxation (11) is that it does not scale well to large estimation problems as the size of the optimization problem scales  $O(m^4)$ . Moreover, the integrality of the solution cannot be guaranteed: Although the constraints are totally unimodular, the objective function is not linear. This problem can be addressed by subsequent projection heuristics. Given the difficulty of the implementation and the fact that it does not even guarantee an improvement over solution at the starting point, we did not pursue this approach in our experiments.

## 4 MULTIVARIATE DEPENDENCE MEASURES

A natural extension is to align several sets of observations. For this purpose, we need to introduce a multivariate version of the Hilbert-Schmidt Independence Criterion. One way of achieving this goal is to compute the Hilbert Space norm of the difference between the expectation operator for the joint distribution and the expectation operator for the product of the marginal distributions since this difference only vanishes whenever the joint distribution and the product of the marginals are identical.

### 4.1 Multivariate Mean Operator

Formally, let there be  $T$  random variables  $x_i \in \mathcal{X}_i$  which are jointly drawn from some distribution  $p(x_1, \dots, x_m)$ . Moreover, denote by  $k_i : \mathcal{X}_i \times \mathcal{X}_i \rightarrow \mathbb{R}$  the corresponding kernels. In this case, we can define a kernel on  $\mathcal{X}_1 \otimes \dots \otimes \mathcal{X}_T$  by

$k_1 \dots k_T$ . The expectation operator with respect to the joint distribution and with respect to the product of the marginals is given by Smola et al. [9]. For instance, the joint expectation operator can be written as follows:

$$\begin{aligned} f(x_1, \dots, x_T) &\rightarrow \mathbf{E}_{x_1, \dots, x_T}[f(x_1, \dots, x_T)] \\ &= \mathbf{E}_{x_1, \dots, x_T}\left[\left\langle f, \prod_{i=1}^T k_i(x_i, \cdot)\right\rangle\right] \\ &= \left\langle f, \mathbf{E}_{x_1, \dots, x_T}\left[\prod_{i=1}^T k_i(x_i, \cdot)\right]\right\rangle. \end{aligned} \quad (12)$$

Hence, we can express the joint expectation operator and the product of the marginal expectation operators in Hilbert space via

$$\mathbf{E}_{x_1, \dots, x_T}\left[\prod_{i=1}^T k_i(x_i, \cdot)\right] \text{ and } \prod_{i=1}^T \mathbf{E}_{x_i}[k_i(x_i, \cdot)], \quad (13)$$

respectively. Straightforward algebra shows that the squared norm of the difference between both terms is given by

$$\begin{aligned} \mathbf{E}_{x_{i=1}^T, x'_{i=1}}\left[\prod_{i=1}^T k_i(x_i, x'_i)\right] &+ \prod_{i=1}^T \mathbf{E}_{x_i, x'_i}[k_i(x_i, x'_i)] \\ &- 2\mathbf{E}_{x_{i=1}^T}\left[\prod_{i=1}^T \mathbf{E}_{x'_i}[k_i(x_i, x'_i)]\right], \end{aligned} \quad (14)$$

which we refer to as multiway HSIC. A biased empirical estimate of the above is obtained by replacing sums by empirical averages. Denote by  $K_i$  the kernel matrix obtained from the kernel  $k_i$  on the set of observations  $X_i := \{x_{i1}, \dots, x_{im}\}$ . In this case, the empirical estimate of (14) is given by

$$\begin{aligned} \text{HSIC}[X_1, \dots, X_T] \\ = 1_m^\top \left( \bigodot_{i=1}^T K_i \right) 1_m + \prod_{i=1}^T 1_m^\top K_i 1_m - 2 \cdot 1_m^\top \left( \bigodot_{i=1}^T K_i \right) 1_m, \end{aligned} \quad (15)$$

where  $\bigodot_{i=1}^T *$  denotes elementwise product of its arguments (the “.\*” notation of MATLAB).

## 4.2 Optimization

To apply this new criterion to sorting, we only need to define  $T$  permutation matrices  $\pi_i \in \Pi_m$  and replace the kernel matrices  $K_i$  by  $\pi_i^\top K_i \pi_i$ .

Without loss of generality, we may set  $\pi_1 = \mathbf{1}$  since we always have the freedom to fix the order of one of the  $T$  sets with respect to which the other sets are to be ordered. In terms of optimization, the same considerations as presented in Section 3 apply. That is, the objective function is convex in the permutation matrices  $\pi_i$  and we may apply DC programming to find a locally optimal solution.

## 5 RELATED WORK

Matching and layout are clearly problems that have attracted a large degree of prior work. We now discuss a number of algorithms which are related to or special cases of what we proposed by means of Kernelized Sorting.

## 5.1 Mutual Information

Probably the most closely related work is that of Jebara [5], who aligns bags of observations by sorting via minimum volume PCA. Here, we show that when using mutual information, our scheme leads to a criterion very similar to the one proposed by Jebara [5]. Mutual information, defined as  $I(X, Y) = h(X) + h(Y) - h(X, Y)$ , is a natural means of studying the dependence between random variables  $x_i$  and  $y_{\pi(i)}$ . In general, this is difficult since it requires density estimation. However, this can be circumvented via an effective approximation, where, instead of maximizing the mutual information directly, we maximize a lower bound to the mutual information. First, we note that only the last term matters since the first two are independent of  $\pi$ . Maximizing a lower bound on the mutual information then corresponds to minimizing an upper bound on the joint entropy  $h(X, Y)$ . An upper bound for the entropy of any distribution with variance  $\Sigma$  is given by the differential entropy of a normal distribution with covariance  $\Sigma$ , which can be computed as

$$h(p) = \frac{1}{2} \log |\Sigma| + \text{constant}. \quad (16)$$

Hence, the problem reduces to minimizing the joint entropy  $J(\pi) := h(\bar{X}, Y)$ , where  $x$  and  $y$  are assumed jointly normal in the reproducing Kernel Hilbert Spaces spanned by the kernels  $k$ ,  $l$ , and  $k \cdot l$ . By defining a joint kernel on  $\mathcal{X} \times \mathcal{Y}$  via  $k((x, y), (x', y')) = k(x, x')l(y, y')$ , we arrive at the optimization problem

$$\underset{\pi \in \Pi_m}{\operatorname{argmin}} \log |HJ(\pi)H| \text{ where } J_{ij} = K_{ij}L_{\pi(i), \pi(j)}. \quad (17)$$

Note that this is related to the optimization criterion proposed by Jebara [5] in the context of sorting via minimum volume PCA. What we have obtained here is an alternative derivation of Jebara’s [5] criterion based on information-theoretic considerations.

The main difference with our work is that Jebara [5] uses the setting to align a large number of bags of observations by optimizing  $\log |HJ(\pi)H|$  with respect to reordering within each of the bags. Obviously, (17) can be extended to multiple random variables simply by taking the pointwise product of a sequence of kernel matrices. In terms of computation, (17) is considerably more expensive to optimize than (5) since it requires computation of inverses of matrices even for gradient computations.

## 5.2 Object Layout

A more direct connection exists between object layout algorithms and Kernelized Sorting. Assume that we would like to position  $m$  objects on the vertices of a graph, such as a layout grid for photographs, with the desire to ensure that related objects can be found in close proximity. We will now show that this is equivalent to Kernelized Sorting between a kernel on objects and the normalized graph Laplacian induced by the graph.

To establish our claim, we need some additional notation. Denote by  $G(V, E)$  an undirected graph with a set of vertices  $V$  and edges  $E$ . With some abuse of notation, we will also denote by  $G$  the symmetric edge adjacency matrix. That is,  $G_{ij} = 1$  if there is an edge between vertices  $i$

and  $j$ , and  $G_{ij} = 0$  if no edge is present. This definition naturally extends to weighted graphs simply by allowing that  $G_{ij} \geq 0$  rather than  $G_{ij} \in \{0, 1\}$ . Moreover, we denote by  $d_i := \sum_j G_{ij}$  the degree of vertex  $i$  in the graph and we let  $D := \text{diag}(d)$  be a diagonal matrix containing the degrees. Finally, we denote by

$$L := D - G \quad (18)$$

the graph Laplacian  $L$ .

It is well known, see, e.g., [21], [22], that local smoothness functionals on graphs can be expressed in terms of  $L$ . More specifically, we have

$$\sum_{i,j} G_{ij} \|\phi(x_i) - \phi(x_j)\|^2 = \text{tr}KL, \quad (19)$$

where  $\phi(x_i)$  can be treated as the vertex value and  $K_{ij} = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ . Basically, (19) sums over the squared differences between the values of adjacent vertices. The smaller the number  $\text{tr}KL$ , the smoother the vertex values vary across the graph. By construction, (19) is translation invariant, that is, changes from  $\phi(x_i) \leftarrow \phi(x_i) - \mu$  leave the functional unchanged. Hence, we have  $\text{tr}KL = \text{tr}HKHL$ .

If we were to lay out objects such that similar objects are assigned to adjacent vertices in  $G$ , we can maximize the smoothness by minimizing  $\text{tr}HKH\pi^\top L\pi$ . Here, the main difference to (5) is that we are *minimizing* a convex form rather than maximizing it.

Such difference can be removed by a simple substitution of  $L$  by  $\|L\|I - L$ . Indeed, note that the eigenvalues of  $L$  range between 0 and  $\|L\|$ . The transformation  $\|L\|I - L$  shifts the eigenvalues into positive territory while changing the objective function only by a constant independent of  $\pi$ , thus leading to a Kernelized Sorting problem for the “kernel”  $L' = \|L\|I - L$ . This is also consistent with the definition of a kernel which is the *inverse* of a regularization operator [23], [24]. That is, while, in a regularization operator, large eigenvalues correspond to properties of a function which are undesirable, the converse is true in a kernel, where large eigenvalues correspond to simple functions [25].

A consequence of these considerations is that for object layout, there exists an alternative strategy for optimization: First, relax the set of permutation matrices  $\Pi_m$  into the set of doubly stochastic matrices  $P_m$  and solve the relaxed problem  $\min_{\pi \in P_m} \text{tr}HKH\pi^\top L\pi$  exactly, and then employ the DC programming described in Section 3.1 to find a locally optimal integral solution. Although theoretically appealing, this approach nonetheless suffers from a range of problems: The number of variables required to deal with in the quadratic program is  $O(m^2)$ , which makes an efficient implementation a challenge even for modest amounts of data unless the special structure of the quadratic form in  $\pi$  is exploited.

### 5.3 Morphing

In object morphing, one may use a compatibility function defined on local similarity between source and destination matches. Assume that  $X, Y \in \mathbb{R}$  are sets of scalars (e.g.,

intensity values in an image). In this context, [7], [8] use scoring functions of the form

$$\frac{1}{2} \sum_{i=1}^m (x_i - y_{\pi(i)})^2 = \frac{1}{2} \sum_i x_i^2 + y_{\pi(i)}^2 - \sum_i x_i y_{\pi(i)}. \quad (20)$$

Whenever  $\pi$  is a bijection,<sup>2</sup> the first two terms are independent of  $\pi$  and the problem of matching becomes one of maximizing  $\sum_i x_i y_{\pi(i)}$ , i.e.,  $X^\top \pi Y$ . By the same argument as in the proof of Lemma 3, this can be rewritten in the form of

$$\underset{\pi \in \Pi_m}{\operatorname{argmax}} \text{tr}XX^\top \pi YY^\top \pi^\top \quad (21)$$

simply by squaring the objective function of  $X^\top \pi Y$ . The only ambiguity left is that of an arbitrary sign, i.e., we might end up *minimizing* the match between  $X$  and  $Y$  rather than maximizing it. This means our argument shows that morphing and Kernelized Sorting have closely related objective functions.

### 5.4 Smooth Collages

The generation of collages is a popular application in the processing of composite images. In this process, one uses a template image  $Y$  (often a company logo or a face of a person) and a collection  $X = \{x_1, \dots, x_m\}$  of reference images to generate a collage where the individual “pixels” of the collage are taken from the set of reference images such that the collage best resembles the template. This problem is easily solved by a linear assignment algorithm as follows.

Denote by  $d(x, y)$  a distance function between an image  $x$  and a pixel  $y$  in the template. Moreover, denote by  $y_i$  a pixel in  $Y$ . In this case, the optimal assignment of reference images to  $Y$  is achieved by finding the permutation  $\pi$  which minimizes

$$\sum_i d(x_i, y_{\pi(i)}) = \text{tr} \pi^\top D \text{ where } D_{ij} := d(x_i, y_j). \quad (22)$$

In other words, one attempts to find an overall allocation of reference images to the template such that the sum of distances is minimized. While this is desirable in itself, it would also be best if there were some spatial coherence between images. This is achieved by mixing the objective function of (22) with the Kernelized Sorting objective. Since this constitutes only a linear offset of the optimization problem of (5), it can be solved in an identical way to what is required in Kernelized Sorting, namely by a DC programming procedure.

## 6 APPLICATIONS

To investigate the performance of our algorithm (it is a fairly nonstandard unsupervised method), we applied it to a variety of different problems ranging from visualization to matching and estimation.

In all of our experiments, the maximum number of iterations used in the updates of  $\pi$  is 100 and we terminate early if progress is less than 0.001 percent of the objective function.

<sup>2</sup> Note that this is *not* required by [7], [8]. In fact, their objective function is not even symmetric between source and destination images.



Fig. 1. Image layouting on a 2D grid and letter grid with kernelized sorting. One can see that images are laid out in the grids according to their color grading. (a) Layout of 320 images into a 2D grid of size  $16 \times 20$  using kernelized sorting. (b) Layout of 280 images into a “PAMI 2009” letter grid using kernelized sorting.

## 6.1 Data Visualization

In many cases, we may want to visualize data according to the metric structure inherent in it. In particular, we may want to align it according to a given template, such as a grid, a torus, or any other *fixed* structure. Such problems occur when presenting images or documents to a user.

While there are a large number of algorithms for low-dimensional object layout (Maximum Variance Unfolding (MVU) [26], Local-Linear Embedding (LLE) [27], ...), most of them suffer from the problem that the low-dimensional presentation is nonuniform. This has the advantage of revealing cluster structure, but, given limited screen size, the presentation is undesirable.

Alternatively, one can use the Self-Organizing Map (SOM) [28] or the Generative Topographic Mapping (GTM) [29] to lay out images according to a predefined grid structure. These methods, however, often map several images into a single grid element, and hence, some grid elements may have no data associated with them. Such grouping creates blank spaces in the layout and still underutilizes the screen space.

Instead, we may use Kernelized Sorting to layout objects. Here, the kernel matrix  $K$  is given by the similarity measure between the objects  $x_i$  that are to be laid out. The kernel  $L$ , on the other hand, denotes the similarity between the locations of grid elements where objects are also to be aligned.

### 6.1.1 Image Layout on a Uniform Grid

For the first visualization experiment, we want to lay out images on a 2D rectangular grid. We have obtained 320 images from Flickr,<sup>3</sup> which are resized and down-sampled to  $40 \times 40$  pixels. We convert the images from RGB into Lab space, yielding  $40 \times 40 \times 3$  dimensional objects. The grid corresponding to  $Y$  is a  $16 \times 20$  mesh on which the images are to be laid out. We use a Gaussian RBF kernel between the objects to be laid out and also between the positions of the grid, i.e.,  $k(x, x') = \exp(-\gamma \|x - x'\|^2)$ . The

kernel width  $\gamma$  is adjusted to the inverse median of  $\|x - x'\|^2$  such that the argument of the exponential is  $O(1)$ . After sorting, we display the images according to their matching coordinates. The result is shown in Fig. 1a. Clearly, images with similar color composition are found at proximal locations.

For comparison, we apply an SOM<sup>4</sup> and a GTM<sup>5</sup> to the same data set. The results are shown in Figs. 2a and 2b. If a grid element (corresponding to a neuron and a latent variable) has been assigned multiple images, only one of the assigned images is displayed. The detail of all other overlapping images can be found on our Web site.<sup>6</sup>

### 6.1.2 Image Layout on an Irregular Grid

To reinforce the point that matching can occur between arbitrary pairs of objects, we demonstrate that images can be aligned with the letters “PAMI 2009” displayed as a pixelated grid on which the images are to be laid out. The same color features and the same Gaussian RBF kernels as in the previous experiment are used. The result is presented in Fig. 1b. As expected, the layout achieves a dual goal: It fully utilizes the elements on the irregular grid while at the same time preserving the color grading.

### 6.1.3 Visualization of Semantic Structure

While color-based image layout gives visually pleasing results, one might desire to lay out images based on their semantic content and explore the high-dimensional semantic space inherent in images by providing a 2D layout. To this end, we represent images as bag of visual words [30], i.e., histograms of vector-quantized local image descriptors. This representation has been shown to be successful in the context of visual object recognition. Here, we use a combination of densely sampled, overlapping patches with

4. <http://www.cis.hut.fi/projects/somtoolbox/>. A Gaussian neighborhood and inverse learning rate functions are used.

5. <http://www.ncrg.aston.ac.uk/GTM/>, again forcing the images into a 2D grid. The principal components are used for the initialization, and the mode projection is used to map data into the (2D grid) latent space.

6. <http://users.rsise.anu.edu.au/~nquadrianto/extras.pdf>.



Fig. 2. Comparison with SOM and GTM for image layout on a 2D grid and a compressed representation of images. Note that both algorithms do not guarantee unique assignments of images to nodes. (a) Layout of 320 images into a 2D grid of size  $16 \times 20$  using SOM. (b) Layout of 320 images into a 2D grid of size  $16 \times 20$  using GTM.

the SIFT descriptor [31]. Then, the inverse of the exponentiated  $\chi^2$  distance, denoted as  $\exp(-\gamma \|x - x'\|_{\chi^2}^2)$ , is used to measure the similarity between the images. Gaussian RBF kernel is still used to measure similarity between the positions of the grid. We apply this scheme to 570 images from the MSRC2 database.<sup>7</sup> The result is presented in Fig. 4.

First, one can observe that objects are grouped according to their categories. For example, books, cars, planes, and people have all or most of their instances visualized in proximal locations. Second, beyond categories, another ordering based on the overall composition of the images is also visible. Images near the lower left corner consist mostly of rectangular-shaped objects; along the antidiagonal direction of the layout, the shapes of the objects become more and more irregular. This reveals structure of the metric space which has not been explicitly designed.

#### 6.1.4 Photo Album Summarization

An immediately useful application of Kernelized Sorting is a tool for presenting a summary of personal photo collections. This is particularly challenging when photos are taken by different people, with different scenery, with different cameras, or over a large time period.<sup>8</sup>

Depending on the way a viewer wants to explore the photo album, the photos can be summarized either based on color information or on a bag-of-visual-words-based image representation. Fig. 5 shows the corresponding summaries for a collection of holiday photos from one of the authors using Kernelized Sorting. Comparing the two summaries, we can see that the latter presents a much clearer separation between natural scenery and human subjects.

## 6.2 Matching

Apart from visualization, Kernelized Sorting can also be used to align or match two related data sets even without cross

data set comparison. In the following set of experiments, we will use data sets with known ground truth of matching. This allows us to quantitatively evaluate Kernelized Sorting. To create such data sets, we either split an image or a vector of data attributes into two halves, or use multilingual documents that are translations of each other.

### 6.2.1 Image Matching

Our first experiment is to match image halves. For this purpose, we use the same set of Flickr images as in Section 6.1.1, but split each image ( $40 \times 40$  pixels) into two equal halves ( $20 \times 40$  pixels). The aim is to match the image halves using Kernelized Sorting. More specifically, given  $x_i$  being the left half of an image and  $y_i$  being the right half of the same image, we want to find a permutation  $\pi$  which lines up  $x_i$  and  $y_{\pi(i)}$  by maximizing the dependence.

Of course, this would be relatively easy if we were allowed to compare the two image halves  $x_i$  and  $y_{\pi(i)}$  directly. While such comparison is clearly feasible for images where we know the compatibility function, it may not be possible for generic objects. Fig. 3 shows the image matching result. For a total of 320 images, we correctly match 140 pairs. This is quite respectable given that the chance level would be only one correct pair (a random permutation matrix has on expectation one nonzero diagonal entry).

### 6.2.2 Multilingual Document Matching

To illustrate that Kernelized Sorting is able to recover nontrivial similarity relations, we apply our algorithm to the matching of multilingual documents in this second experiment. For this purpose, we use the Europarl Parallel Corpus.<sup>9</sup> It is a collection of the proceedings of the European Parliament, dating back to 1996 [32]. We select the 300 longest documents of Danish (Da), Dutch (Nl), English (En), French (Fr), German (De), Italian (It), Portuguese (Pt), Spanish (Es),

7. <http://research.microsoft.com/vision/cambridge/recognition/>.

8. In fact, the photos are taken from a collection of holiday photographs of one of the authors—without consideration of using them for the purpose of this paper. The equipment was a consumer grade point-and-shoot camera and a digital SLR with high-quality lenses. Two users took the pictures.

9. <http://www.statmt.org/europarl/>.



Fig. 3. Image matching as obtained by kernelized sorting. The images are cut vertically into two equal halves and kernelized sorting is used to pair up image halves that originate from the same images.

and Swedish (Sv). The purpose is to match the non-English documents (source languages) to their English translations (target language). Note that our algorithm does *not* require a cross-language dictionary. In fact, one could use Kernelized Sorting to generate a dictionary after an initial matching has been created.

We use standard TF-IDF (term frequency-inverse document frequency) features of a bag-of-words kernel. As preprocessing, we remove stopwords (via NLTK<sup>10</sup>) and perform stemming using Snowball.<sup>11</sup> Finally, the feature vectors are normalized to unit length in term of  $\ell_2$  norm. Since these kernel matrices on documents are notoriously diagonally dominant, we use the bias-corrected version of our optimization problem.

As a reference, we use a fairly straightforward means of document matching via its length. That is, longer documents in one language will be most probably translated into longer documents in the other language. This observation has also been used in the widely adopted sentence alignment method [33]. Alternatively, we can use a dictionary-based method as an upper bound for what can be achieved by matching. We translate the documents in the source languages into the target language word by word using Google Translate.<sup>12</sup> This effectively allows us to directly compare documents written in different languages. Now, for each source language and the target language, we can compute a kernel matrix based on a bag-of-words kernel, and the  $ij$ th entry of this kernel matrix is the similarity between document  $i$  in the source language and document  $j$  in the target language. Then, we can use this kernel matrix and a linear assignment to find the matches between documents across languages.

The experimental results are summarized in Table 1. Here, we use two versions of our algorithm: one with a fixed set of  $\lambda$ s and the other with automatic tuning of  $\lambda$  (as in Section 3). In practice, we find that trying out different  $\lambda$

10. <http://nltk.sf.net/>.

11. <http://snowball.tartarus.org>.

12. <http://translate.google.com>. Note that we did not perform stemming on the words, and thus, the dictionary is highly customized to the problem at hand.

TABLE 1  
The Number of Correct Matches from Documents Written in Various Source Languages to Those in English

Language	PT	ES	FR	SV	DA	IT	NL	DE
KS	252	218	246	150	230	237	223	95
KS - LS	241	216	193	99	83	236	211	70
RE	9	12	8	6	6	11	7	4
UB	298	298	298	296	297	300	298	284

We compare Kernelized Sorting (KS) to a reference procedure which simply matches the lengths of documents (RE: Reference) and a dictionary-based approach (UB: Upper Bound). We also include results of line search or automatic tuning of  $\lambda$  (KS-LS). Reported are the numbers of correct matches (out of 300) for various source languages.

from a fixed set ( $\lambda \in \{0.1, 0.2, \dots, 1.0\}$ ) and then choosing the best  $\lambda$  in terms of the objective function works better than automatic tuning. Low matching performance for the document-length-based method might be due to small variance in the document length after we choose the 300 longest documents. The dictionary-based method gives near-perfect matching. Our method produces results consistent with the dictionary-based method, for instance, the notably low performance for matching German documents to its English translations. We suspect that the difficulty of German-English document matching is inherent to this data set as it was also observed in [32]. Arguably, the matching produced by Kernelized Sorting is quite encouraging as our method uses only a within-language similarity measure while still matching more than 2/3 what a dictionary-based method is capable of in most cases.

### 6.2.3 Data Attribute Matching

In our last experiment, we aim to match attributes of vectorial data. In our setup, we use benchmark data sets for supervised learning from the UCI repository<sup>13</sup> and LibSVM site.<sup>14</sup> We split the attributes (or dimensions) of each data point into two halves, and we want to match them back. Here, we use the estimation error to quantify the quality of the match. That is, assumed that  $y_i$  is associated with the observation  $x_i$ . In this case, we compare  $y_i$  and  $y_{\pi(i)}$  using homogenous misclassification loss for binary and multiclass problems and squared loss for regression problem. Note that this measure of goodness is different from the ones we used in image matching and document matching. This is because, for data attribute matching, we may not be able to match back the two halves of an individual data point exactly, but we can restore the overall characteristic of the data such as class separability.

To ensure good dependence between the splitted attributes, we choose a split which ensures correlation. This is achieved as follows: First, we compute the correlation matrix of the data; then, among the pairs of attributes which achieves the largest correlation, we pick the dimension with the smallest index as the reference; next, we choose the dimensions that have at least 0.5 correlation with the reference and split them equally into two sets, set A and set B (we also put the reference dimension into set A); last, we divide the remaining dimensions (with less than 0.5 correlation with the reference) into two equal halves,

13. <http://archive.ics.uci.edu/ml>.

14. <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools>.



Fig. 4. Layout of 570 images into a 2D grid of size  $15 \times 38$  using bag-of-visual-words-based kernelized sorting. Several object categories, like books, cars, planes, and people, are grouped into proximal locations.

and allocate them into sets A and B, respectively. This scheme ensures that at least one dimension in set B is strongly correlated with at least one dimension in set A. The detailed split of the data attributes for different data sets can be found on our Web site.<sup>15</sup>

As before, we use a Gaussian RBF kernel with median adjustment for the kernel width for both  $x$  and  $y$ . To obtain statistically meaningful results, we subsample 80 percent of the data 10 times and compute the error of the match on the subset (this is done in lieu of cross-validation since the latter is meaningless for matching). As a reference, we compute the expected performance of random permutations which can be done exactly.<sup>16</sup> As a lower bound for the estimation error, we use the original data set and perform classification/regression using 10-fold cross validation. The results are summarized in Table 2. Basically, the closer the results obtained by Kernelized Sorting to the lower bound, the better. In many cases, Kernelized Sorting is able to restore significant information related to the class separability in the classification problems and the functional relationship in the regression problems.

### 6.3 Multivariate Extension

In this experiment, we align five USPS digits of 0s using multiway HSIC. In this case, each nonzero pixel in an image is a data point and each image has 100 nonzero pixels. On each set of digits, we use a Gaussian RBF kernel with median adjustment of the kernel width. Furthermore, we use the first digit as the target set (i.e.,  $\pi_1 = I$ ) and the other digits as the sources. The sorting performance is visualized by computing linear interpolations between the matching pixels. If meaningful matching is obtained, such interpolation will result in meaningful intermediate images [5].

For comparison, we also perform the same task using the method proposed by Jebara [5]. Briefly, Jebara [5] proposes a method to sort many sets (or bags) of objects by

maximizing likelihood under a Gaussian model to minimize the volume data occupied in Hibert space. An iterative likelihood maximization procedure is devised by interleaving update of Gaussian's moments and adjustment of permutation configuration of each set of objects. We implemented our own version as we were unable to obtain their code for reasons beyond the control of Jebara [5]. We only experimented with the simpler version using the mean estimator (locking covariance matrix as a constant multiplication of an identity matrix) and LAP as it was observed that this simpler version performs as well as his more sophisticated counterpart based on a covariance estimator (allowing covariance matrix as an arbitrary positive semi-definite matrix) [5]. Here, we also use a Gaussian RBF kernel with median trick as the base kernel. Although we are only interested in sorting five digits of 0s, the method of Jebara [5] requires more digits (200 in our experiments) to get a decent ML estimate of the feature space mean. As such, the usage of Jebara's [5] method in finding a correspondence with just two sets of observations (as in Sections 6.1, 6.2.1, 6.2.2, and 6.2.3), i.e., this translates to get an ML mean estimate of the Gaussian likelihood with just two samples, is not obvious.

The interpolation results are shown in Figs. 6a and 6b. Because of the symmetric structure of the 0s digit, some of

TABLE 2  
Estimation Error for Data Attribute Matching

Type	Data set	$m$	KS	RE	LB
Binary	australian	690	$0.29 \pm 0.02$	0.49	$0.21 \pm 0.04$
	breastcancer	683	$0.06 \pm 0.01$	0.46	$0.06 \pm 0.03$
	derm	358	$0.08 \pm 0.01$	0.43	$0.00 \pm 0.00$
	optdigits	765	$0.01 \pm 0.00$	0.49	$0.01 \pm 0.00$
	wdbc	569	$0.11 \pm 0.04$	0.47	$0.05 \pm 0.02$
Multiclass	satimage	620	$0.20 \pm 0.01$	0.80	$0.13 \pm 0.04$
	segment	693	$0.58 \pm 0.02$	0.86	$0.05 \pm 0.02$
	vehicle	423	$0.58 \pm 0.08$	0.75	$0.24 \pm 0.07$
Regression	abalone	417	$13.9 \pm 1.70$	18.7	$6.44 \pm 3.14$
	bodyfat	252	$4.5 \pm 0.37$	7.20	$3.80 \pm 0.76$

We compare estimation errors between the original data set (LB: Lower Bound), data set after KS, and data set after random permutation (RE: Reference).

15. <http://users.rsise.anu.edu.au/~nquadrianto/extras.pdf>.

16. For classification:  $1 - \sum_{i=1}^{|Y|} p_i^2$  and for regression:  $2(\mathbf{E}_y[y^2] - \mathbf{E}_y^2[y])$ . Here,  $y$  denotes the class label and  $p_i$  denotes the proportion of class  $i$  in the data set.

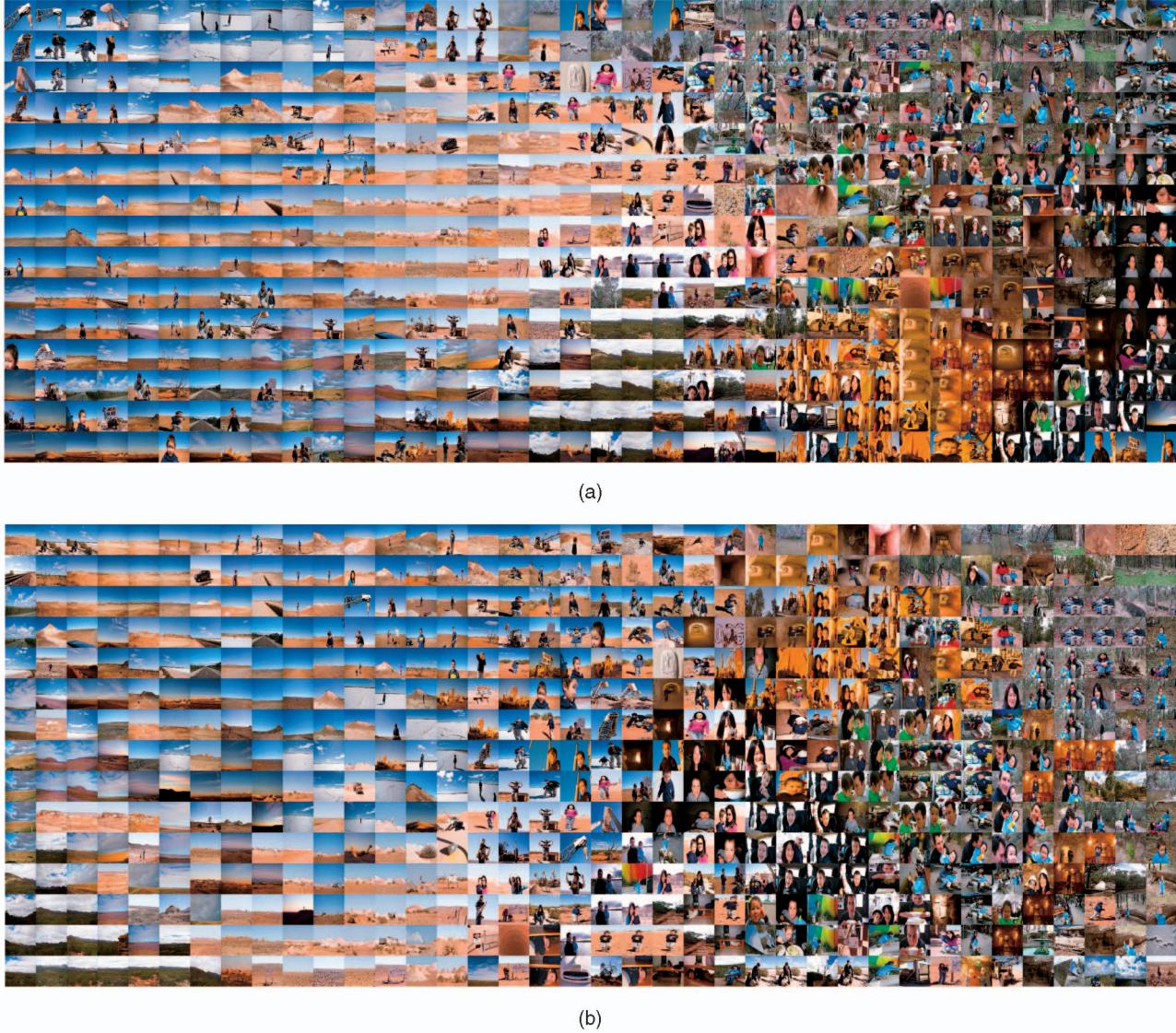


Fig. 5. Application of kernelized sorting as a photo collection summarization tool. (a) Photos summarization by color-based kernelized sorting. (b) Photos summarization by bag-of-visual-words-based kernelized sorting.

the correspondences are reversed (the top is matched to the bottom and the bottom is matched to the top), which is apparent from Fig. 6a. Nevertheless, the interpolations obtained with HSIC seem to produce a better local consistency than those obtained with entropy. This is clear from the flows of arrows in the velocity plots (arrows are pointing away from a matching pixel in the source digits) shown in Figs. 6c and 6d for each digit pair shown in Figs. 6a and 6b. For example, in the top right plot of Fig. 6c, all of the arrows “inside” the 0 are pointing downward. However, in Fig. 6d, some arrows are pointing downward, but some upward. This local flow consistency implies that in the matching, neighboring pixels in one digit will be mapped to the neighboring locations in the other digit as well.

## 7 CONCLUSION

In this paper, we generalized sorting by maximizing the dependency between matched pairs of observations by means of the Hilbert-Schmidt Independence Criterion. This

way, we are able to perform matching *without* the need of a cross-domain similarity measure, and we managed to put sorting and assignment operations onto an information-theoretic footing. The proposed sorting algorithm is efficient, and it can be applied to a variety of different problems ranging from data visualization to image and multilingual document matching. Moreover, we showed that our approach is closely related to matching and object layout algorithms, and that by changing the dependence measure, we are able to recover previous work on sorting in Hilbert Spaces.

## ACKNOWLEDGMENTS

The authors thank Marconi Barbosa, Tiberio Caetano and the anonymous reviewers for detailed comments. NICTA is funded through the Australian Government’s *Backing Australia’s Ability* initiative, in part through the ARC. This research was supported by the Pascal Network. Parts of this work were done while Alex J. Smola, Le Song, and Tinne Tuytelaars were working at NICTA. Tinne Tuytelaars is

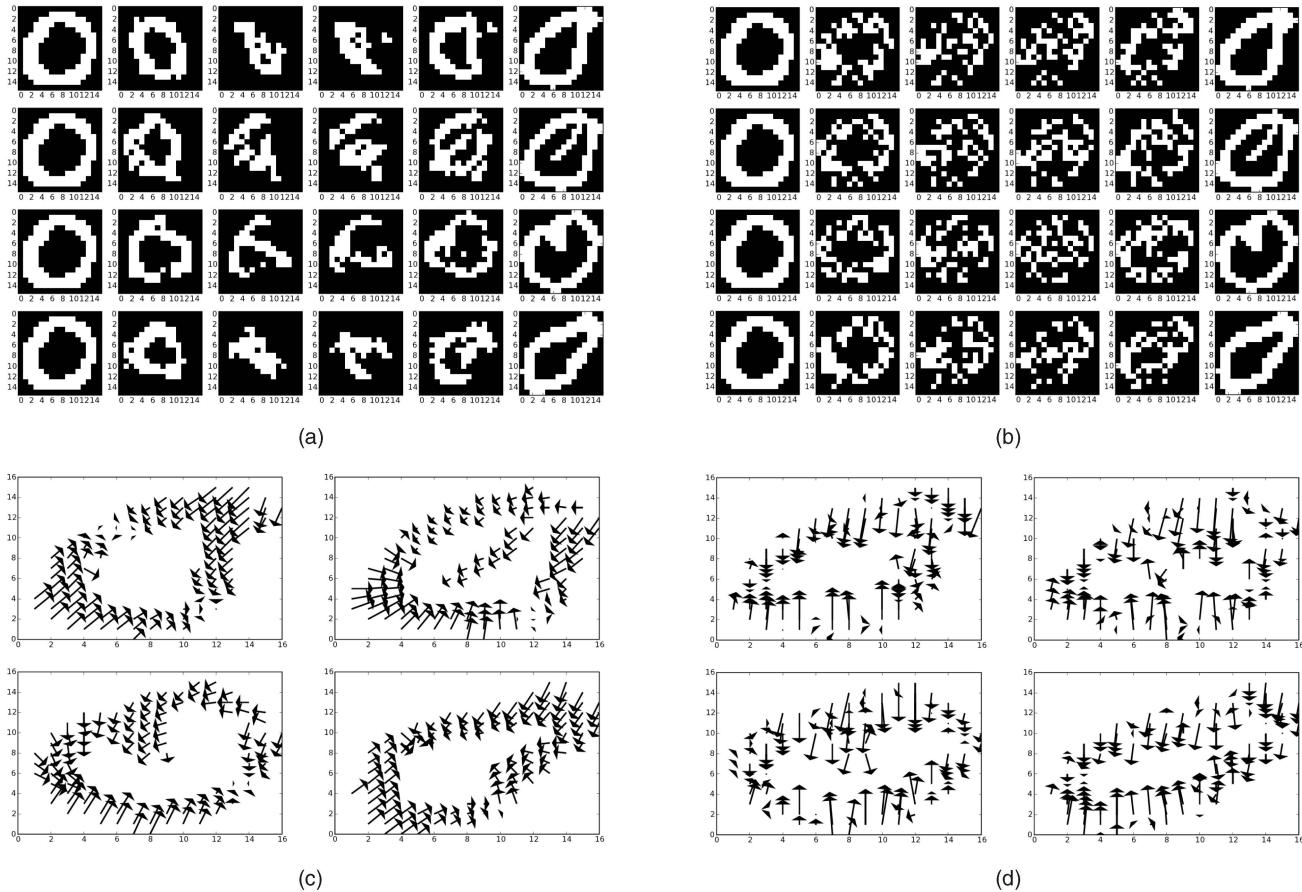


Fig. 6. Linear interpolation of four pairs of the digit 0 after sorting using multiway HSIC and entropy [5]. (a) Linear interpolation using multiway HSIC. (b) Linear interpolation using entropy. (c) Arrows showing the matching of strokes of digit pairs sorted using multiway HSIC. (d) Arrows showing the matching of strokes of digit pairs sorted using entropy.

supported by a postdoctoral research grant of the Fund for Scientific Research Flanders. A short version of this paper appeared in [1].

## REFERENCES

- [1] N. Quadrianto, L. Song, and A. Smola, "Kernelized Sorting," *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., vol. 21, pp. 1289-1296, MIT Press, 2009.
- [2] S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 377-388, Apr. 1996.
- [3] T. Caetano, L. Cheng, Q.V. Le, and A.J. Smola, "Learning Graph Matching," *Proc. 11th IEEE Int'l Conf. Computer Vision*, pp. 1-8, 2007.
- [4] T. Cour, P. Srinivasan, and J. Shi, "Balanced Graph Matching," *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hofmann, eds., vol. 19, pp. 313-320, MIT Press, Dec. 2006.
- [5] T. Jebara, "Kernelizing Sorting, Permutation, and Alignment for Minimum Volume PCA," *Proc. Conf. Computational Learning Theory*, pp. 609-623 2004.
- [6] T. Caetano, T. Caelli, D. Schuurmans, and D. Barone, "Graphical Models and Point Pattern Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1646-1663, Oct. 2006.
- [7] C. Walder, B. Schölkopf, and O. Chapelle, "Implicit Surface Modelling with a Globally Regularised Basis of Compact Support," *Computer Graphics Forum*, vol. 25, no. 3, pp. 635-644, 2006.
- [8] F. Steinke, B. Schölkopf, and V. Blanz, "Learning Dense 3D Correspondence," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pp. 1313-1320, 2007.
- [9] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A Hilbert Space Embedding for Distributions," *Algorithmic Learning Theory*, E. Takimoto, ed., Springer, 2007.
- [10] B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf, "Injective Hilbert Space Embeddings of Probability Measures," *Proc. 21st Ann. Conf. Learning Theory*, pp. 111-122, 2008.
- [11] N. Aronszajn, "La Théorie Générale des Noyaux Réproduisants et ses Applications," *Proc. Cambridge Philosophical Soc.*, vol. 39, pp. 133-153, 1944.
- [12] K. Fukumizu, F.R. Bach, and M.I. Jordan, "Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces," *J. Machine Learning Research*, vol. 5, pp. 73-99, 2004.
- [13] G. Finke, R.E. Burkard, and F. Rendl, "Quadratic Assignment Problems," *Annals of Discrete Math.*, vol. 31, pp. 61-82, 1987.
- [14] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
- [15] S. Sherman, "On a Theorem of Hardy, Littlewood, Polya, and Blackwell," *Proc. Nat'l Academy of Sciences*, vol. 37, pp. 826-831, 1951.
- [16] C. McDiarmid, "On the Method of Bounded Differences," *Survey in Combinatorics*, pp. 148-188, Cambridge Univ. Press, 1989.
- [17] T.P. Dinh and L.H. An, "A D.C. Optimization Algorithm for Solving the Trust-Region SubProblem," *SIAM J. Optimization*, vol. 8, no. 2, pp. 476-505, 1998.
- [18] A. Yuille and A. Rangarajan, "The Concave-Convex Procedure," *Neural Computation*, vol. 15, pp. 915-936, 2003.
- [19] R. Jonker and A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems," *Computing*, vol. 38, pp. 325-340, 1987.
- [20] W. Gander, G. Golub, and U. von Matt, "A Constrained Eigenvalue Problem," *Linear Algebra and Its Applications*, vol. 114-115, pp. 815-839, 1989.

- [21] M. Fiedler, "Algebraic Connectivity of Graphs," *Czechoslovak Math. J.*, vol. 23, no. 98, pp. 298-305, 1973.
- [22] F. Chung-Graham, *Spectral Graph Theory*. AMS, 1997.
- [23] F. Girosi, "An Equivalence between Sparse Approximation and Support Vector Machines," A.I. Memo no. 1606, Artificial Intelligence Laboratory, Massachusetts Inst. of Technology, 1997.
- [24] A.J. Smola, B. Schölkopf, and K.-R. Müller, "The Connection between Regularization Operators and Support Vector Kernels," *Neural Networks*, vol. 11, no. 5, pp. 637-649, 1998.
- [25] B. Schölkopf, A.J. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, pp. 1299-1319, 1998.
- [26] K.Q. Weinberger and L.K. Saul, "An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding," *Proc. Nat'l Conf. Artificial Intelligence*, 2006.
- [27] S. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, pp. 2323-2326, Dec. 2000.
- [28] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.
- [29] C.M. Bishop, M. Svensén, and C.K.I. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation*, vol. 10, no. 1, pp. 215-234, 1998.
- [30] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," *Proc. Workshop Statistical Learning in Computer Vision*, pp. 1-22, 2004.
- [31] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, pp. 91-110, 2004.
- [32] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," *Proc. Machine Translation Summit X*, pp. 79-86, 2005.
- [33] W.A. Gale and K.W. Church, "A Program for Aligning Sentences in Bilingual Corpora," *Proc. Meeting of the Assoc. for Computational Linguistics*, pp. 177-184, 1991.



**Le Song** received the bachelor's degree in computer science from South China University of Technology, Guangzhou, in 2002, and the master's and doctoral degrees in computer science from the University of Sydney, Australia, in 2004 and 2008, respectively. He was with the Statistical Machine Learning Program at NICTA. Currently, he is a Lane Fellow at Carnegie Mellon University. He is affiliated with the Machine Learning Department as well as the Lane Center

for Computational Biology. His research interests include Hilbert space embedding of distributions, kernel methods, exponential families, statistical modeling, and analysis of networks. He also works on applications in bioinformatics, social networks, computer vision, and document analysis.



**Tinne Tuytelaars** received the master of electrical engineering degree and the PhD degree from the Katholieke Universiteit Leuven (K.U. Leuven), Belgium, in 1996 and 2000, respectively. Since October 2008, she has been a research professor (BOF-ZAP) at K.U. Leuven, where she has mostly been working in the VISICS-Lab within ESAT-PSI. Her main research interests include image representations, object recognition, and multimodal analysis (combining images and text). She has published more than 60 papers in peer-reviewed conferences and journals. She regularly serves as a program committee member or area chair for the major computer vision conferences ICCV, ECCV, and CVPR, and regularly reviews for the major computer vision journals *International Journal of Computer Vision* and *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).



**Novi Quadrianto** received the BEng degree in electrical and electronic engineering (with first class honors) from Nanyang Technological University, Singapore, in 2005, and the master's degree in computer science in 2007 from the Australian National University, where he is currently working toward the PhD degree. From 2005 to 2006, he was a research officer at the Institute for Infocomm Research, Singapore. He was awarded a silver prize for his undergraduate project in brain-computer interface. His research interests include statistical machine learning, particularly kernel methods and exponential families, statistical signal processing, and neural signal processing.



**Alex J. Smola** received the master's degree in physics from the University of Technology, Munich, and the doctoral degree in computer science from the University of Technology, Berlin. Until 1999, he was a researcher at the IDA Group, GMD Institute for Software Engineering and Computer Architecture in Berlin (now part of the Fraunhofer Gesellschaft). He was a researcher and a group leader at the Research School for Information Sciences and Engineering, Australian National University. From 2004 to 2008, he was a senior principal researcher and the program leader of the Statistical Machine Learning Group at NICTA. He is now at Yahoo! Research. He has published more than 100 papers, written one book, and edited five books. His specialties are kernel methods, such as support vector machines and Gaussian processes. He has served on several senior program committees for NIPS, COLT, and ICML. He is a member of the editorial boards of the *Journal of Machine Learning Research*, *Statistics and Computing*, and the *IEEE Transactions on Pattern Analysis and Machine Intelligence*.