The Multi-Task Learning View of Multimodal Data

Hachem Kadri HACHEM.KADRI@LIF.UNIV-MRS.FR
Stéphane Ayache STEPHANE.AYACHE@LIF.UNIV-MRS.FR
Cécile Capponi CECILE.CAPPONI@LIF.UNIV-MRS.FR
Sokol Koço SOKOL.KOCO@LIF.UNIV-MRS.FR
François-Xavier Dupé FRANCOIS-XAVIER.DUPE@LIF.UNIV-MRS.FR
Emilie Morvant EMILIE.MORVANT@LIF.UNIV-MRS.FR

Aix-Marseille Université, CNRS, LIF UMR 7279, 13000, Marseille, France

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

We study the problem of learning from multiple views using kernel methods in a supervised setting. We approach this problem from a multi-task learning point of view and illustrate how to capture the interesting multimodal structure of the data using multi-task kernels. Our analysis shows that the multi-task perspective offers the flexibility to design more efficient multiple-source learning algorithms, and hence the ability to exploit multiple descriptions of the data. In particular, we formulate the multimodal learning framework using vector-valued reproducing kernel Hilbert spaces, and we derive specific multi-task kernels that can operate over multiple modalities. Finally, we analyze the vector-valued regularized least squares algorithm in this context, and demonstrate its potential in a series of experiments with a real-world multimodal data set.

Keywords: multimodal learning, multiple views, multi-task kernels, vector-valued RKHS, cross-covariance operator.

1. Introduction

Learning from multiple sources[1], a.k.a multi-view learning, deals with the integration of different representations of the data, that can be either redundant, complementary, independent, or contradictory, in order to solve a learning problem (Cesa-Bianchi et al., 2010). Taking into account multimodal inputs can bring various information useful for improving the quality of the learned classifier. This could be very helpful in applications such as bioinformatics, computer vision, multimedia indexing and web search, where several descriptions on the same objects are available.

A number of studies have appeared in the last decade on multi-view learning, which can be broadly classified into three categories:

[1] Sources, or views, or modalities: the term multi-view learning is used to describe all learning algorithms that exploit different views of the input data, whether these views are independent or not.

- *supervised setting*: multi-view learning is achieved through early fusion of descriptors then learning, or by learning on each view followed by a late fusion (Muslea and Knoblock, 2006; Kludas et al., 2007);

- *semi-supervised setting*: starting from co-training (Blum and Mitchell, 1998), learning approaches that fall in this category aim at exploiting the agreements between views, usually through regularization approaches on the non-labeled examples (Balcan et al., 2004; Christoudias et al., 2008; Sindhwani and Rosenberg, 2008; Wang and Zhou, 2008);

- *unsupervised setting*: the idea here is to apply the co-regularization framework to unsupervised learning problems, such as clustering and dimensionality reduction (Kumar and Daumé III, 2011; Kumar et al., 2011).

In this paper, we focus on the supervised learning part. We tackle the multi-view learning problem with a multi-task learning point of view and illustrate how to capture the interesting multimodal structure of the data using multi-task kernels. Indeed, learning from multiple views is closely related to multi-task learning, and some work has pointed out this connection (Szedmak et al., 2005; Cavallanti et al., 2008; Crammer et al., 2008). However, most research in the area of multi-view (or/and multimodal) learning seems to have evolved as a "parallel learning strategy" to the larger area of multi-task learning. While the two learning mechanisms are similar, the viewpoints and ways of thinking about these tasks tend to be quite different. A major goal of this paper is to make some directly ideas from multi-task learning applicable in multi-view setting.

Learning from multiple views can be viewed as a special case of multi-task learning, where all tasks share the same label (Cavallanti et al., 2008). To further strengthen the link between multi-task and multi-view learning in this paper, we take advantage of the flexibility endowed by the vector-valued reproducing kernel Hilbert space (RKHS) framework (Micchelli and Pontil, 2005) and the ability of multi-task kernels (Micchelli and Pontil) to operate over multiple modalities (Section 3). Then, we describe and analyze the vector-valued regularized least squares algorithm for learning when multiple views of the data are provided for training (Section 4). Finally, we demonstrate the merits of our approach on a real-world multimodal data set (Section 5).

To the best of our knowledge, the most directly related work to this paper is that of Luo et al. (2013) and Minh et al. (2013). In these two papers, the authors have studied the problem of multi-view learning using vector-valued RKHS in semi-supervised setting. However, in (Luo et al., 2013) the vector-valued approach has been adopted to deal with multi-label classification and not the multi-view setup. A multiple kernel learning (MKL) approach has been used due to the presence of multiple views of input data, but this fails to capture between-view information. In (Minh et al., 2013) the connection between multi-view learning and vector-valued manifold regularization has been explored, providing a unifying framework linking these two learning approaches. In this setting, a crucial issue, which has not been discussed in (Minh et al., 2013), is the choice of the multi-task kernel as a way to exhibit the interesting multimodal structure of the data. We address this by introducing multi-task kernels based on cross-covariance operators on RKHS that allow modeling variables of multiple types and modalities. The remainder of the paper is mainly concerned with vector-valued RKHS and multi-task kernels, and their connections to supervised learning from multiple sources/views.

2. Multi-Task Learning with Kernels: A Brief Review

In this section, we briefly review the formal setup for multi-task learning using regularization in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with a multi-task kernel $K(\cdot,\cdot)$. It extends the single-task kernel learning framework (Schölkopf and Smola, 2001) to multi-task setting. We refer the reader to (Evgeniou et al., 2005) and references therein for more details.

In the standard multi-task learning setup we have p tasks. For the l-th task, $l \in \mathbb{N}_p$ (we use the notation $\mathbb{N}_p$ for the set $\{1,\dots,p\}$), n examples $\{(x_{il}, y_{il}) : i \in \mathbb{N}_n\} \subset \mathcal{X}_l \times \mathcal{Y}_l$ sampled *i.i.d.* from an unknown probability distribution P_l on $\mathcal{X}_l \times \mathcal{Y}_l$ are available. The total data available is then $\{(x_{il}, y_{il}) : i \in \mathbb{N}_n, l \in \mathbb{N}_p\}$, and the goal is to learn all p functions $f_l : \mathcal{X}_l \to \mathcal{Y}_l$ simultaneously from the available data. It is common in multi-task learning to assume that the tasks share the same input space, that is $\mathcal{X}_l = \mathcal{X}$ for all l. In this case, multi-task learning can be formulated as a vector-valued function learning problem (Micchelli and Pontil, 2005), where the goal is to learn a function f from $\mathcal{X}$ to $\mathcal{Y}$; the input space $\mathcal{X}$ is typically a subset of $\mathbb{R}^d$, the d dimensional Euclidean space, and the output space $\mathcal{Y}$ is a subset of $\mathbb{R}^p$ containing the outputs of the p tasks. To this end, a common approach is to learn f as the minimizer in a vector-valued RKHS $\mathcal{H}$ of the functional

$$\frac{1}{n}\sum_{i\in\mathbb{N}_n} V(y_i, f(x_i)) + \lambda\|f\|^2_{\mathcal{H}}, \tag{1}$$

where $y_i = (y_{il})_{l\in\mathbb{N}_p} \in \mathbb{R}^p$, $\lambda > 0$ is a regularization parameter, and $\|f\|_{\mathcal{H}}$ is the norm of f in $\mathcal{H}$. V is a convex loss function such as the square error $\|y_i - f(x_i)\|^2_{\mathcal{Y}}$.

A vector-valued RKHS is uniquely defined by a positive definite multi-task kernel $K(\cdot,\cdot)$; that is a matrix-valued function of two variables satisfying the following two properties.

Definition 1. *(Positive definite multi-task kernel). A function $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$, where $\mathcal{L}(\mathcal{Y})$ is the the space of bounded linear operators from $\mathcal{Y}$ into itself* [2]*, is positive definite if the following holds*

(a) $K(x,z)^* = K(z,x)$, for any $(x,z) \in \mathcal{X}^2$, where * denotes the adjoint,

(b) $\sum_{i,j\in\mathbb{N}_q} \langle y_i, K(x_i,x_j)y_j\rangle_{\mathcal{Y}} \geq 0$, for any $q \in \mathbb{N}$, $\{x_i : i \in \mathbb{N}_q\} \subseteq \mathcal{X}$, and $\{y_i : i \in \mathbb{N}_q\} \subseteq \mathcal{Y}$.

Corresponding to any such kernel $K(\cdot,\cdot)$ there is a map Φ from $\mathcal{X} \times \mathcal{Y}$ to a *feature space* $\mathcal{F}$, such that:

$$\langle y, K(x,z)v\rangle_{\mathcal{Y}} = \langle \Phi(x,y), \Phi(z,v)\rangle_{\mathcal{F}},$$

for any $(x,z)\in\mathcal{X}^2$ and $(y,v)\in\mathcal{Y}^2$. Hence, the kernel can be used, as with the basic scalar-valued kernel, to compute an inner product in the feature space. This is referred to as the multi-task "kernel trick". Note that the usual kernel trick can be recovered when $p = 1$.

[2] If p, the number of tasks, is finite, $\mathcal{Y} \subseteq \mathbb{R}^p$ is finite-dimensional and $\mathcal{L}(\mathcal{Y}) = \mathbb{R}^{p\times p}$ is the vector space of symmetric matrices of size $p \times p$. In this case, the multi-task kernel $K(\cdot,\cdot)$ is also called matrix-valued kernel (Reisert and Burkhardt, 2007).

One particularly attractive instantiation of such a feature space is the vector-valued RKHS $\mathcal{H}$ associated with the multi-task kernel $K(\cdot,\cdot)$. Consider the set of functions $\{K(\cdot,x)y : x \in \mathcal{X}, y \in \mathcal{Y}\}$, where x and y index the set of functions and the dot represents the argument to a given function. The span of such functions defines a linear space that is unique and can always be completed into a Hilbert space (Schwartz, 1964; Micchelli and Pontil, 2005). The crucial property of these Hilbert spaces is the "reproducing property" of the kernel

$$\langle y, f(x)\rangle_{\mathcal{Y}} = \langle K(\cdot,x)y, f\rangle_{\mathcal{H}},$$

for any $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $f \in \mathcal{H}$. Note in particular that if we define $\Phi(x,y) = K(\cdot,x)y$ as a map into the vector-valued RKHS, then we have

$$\langle \Phi(x,y), \Phi(z,v)\rangle_{\mathcal{F}} = \langle K(\cdot,x)y, K(\cdot,z)v\rangle_{\mathcal{F}} = \langle y, K(x,z)v\rangle_{\mathcal{Y}},$$

and thus $\Phi(x,y) = K(\cdot,x)y$ is indeed an instantiation of the feature space perspective of the multi-task kernel. For completeness, we recall the definition of vector-valued RKHS.

Definition 2. *(vector-valued RKHS). A vector-valued reproducing kernel Hilbert space is a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathcal{Y}$ that possesses a (matrix-valued) multi-task **reproducing** kernel, i.e., a function $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ for which the following holds*

(a) $K(x,\cdot)y \in \mathcal{H}$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$,

(b) $\langle f, K(\cdot,x)y\rangle_{\mathcal{H}} = \langle f(x), y\rangle_{\mathcal{Y}}$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $f \in \mathcal{H}$.

It is important to note that there is a bijection between positive semidefinite multi-task kernels and vector-valued RKHS. Moreover, the multi-task kernel associated with a vector-valued RKHS is unique. This guarantees the unicity of the solution to the problem (1) which is given by the representer theorem (Micchelli and Pontil, 2005).

Representer theorem. *(vector-valued case). Any solution to the problem: find $f \in \mathcal{H}$ to minimize (1) has a representation of the form*

$$f(\cdot) = \sum_{i \in \mathbb{N}_n} K(\cdot, x_i)c_i, \tag{2}$$

where $c_i \in \mathcal{Y}$.

The result in Equation (2) is noteworthy as it makes the optimization problem (1) amenable for computations. In particular, the unique minimizer of functional (1) can be found by replacing f by the right hand side of Equation (2) in Equation (1) and then optimizing with respect to the parameters $\{c_i : i \in \mathbb{N}_n\}$.

We now show how multi-task learning using regularized kernel methods, as described above, is useful for "supervised" learning when multiple views of the data are available.

3. Multi-View Learning from a Multi-Task Perspective

In this section, we start by formalizing the link between kernel-based multi-task learning and multi-view learning. We then discuss multi-task kernels built from cross-covariance operators as a viable way of modeling within- and between-view information.

3.1. Framework

Consider the general setting of supervised learning with multiple views, where we aim to identify a target function given a set of labeled data drawn from m different sources. The training data is composed by multiple and various sets of features. One way to tackle this problem is to define a kernel for each view and to consider a regularized risk minimization problem in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$

$$\underset{\mathbf{f}(\cdot)\in\mathcal{H}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n} V(y_i, \mathbf{f}(x_i)) + \lambda\|\mathbf{f}\|^2, \tag{3}$$

where $\mathbf{f} = (f_1, \dots, f_m) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_m$, $V(y, \cdot)$ is a convex loss function, and $\lambda \geq 0$ is a regularization parameter. Given n training data $\{(x_i, y_i) : i \in \mathbb{N}_n\} \subset (\mathcal{X} \times \mathcal{Y})^n$ (we use the notation $\mathbb{N}_n$ for the set $\{1, \dots, n\}$) where each example $x_i = (x_i^{(1)}, \dots, x_i^{(m)})$ is seen in m views, $\mathcal{H}_1, \dots, \mathcal{H}_m$ are the scalar-valued RKHSs associated respectively to the reproducing kernels $k_1, \dots, k_m$ defined for each view.

Much previous work in the machine learning literature has focused on learning $\mathbf{f}$ in a scalar-valued RKHS (Brefeld et al., 2006; Farquhar et al., 2006; Sindhwani and Rosenberg, 2008). This means that the decision function $\mathbf{f}(x)$ is obtained by computing the m functions f_s, $s \in \mathbb{N}_m$, independently for each view. It is true that these previous work succeed in taking into account dependencies between views ; however, this is achieved only through regularization and mainly in the presence of unlabeled data. In the supervised setting, using a scalar-valued RKHS and learning independently the function $(f_s)_{s\in\mathbb{N}_m}$ may fail to fully take into account the structure of multimodal data, since it is difficult to optimize the agreement between views under some assumptions (redundancy of views and conditional independence) or to promote some other properties among views, especially when the number of views m is large. Vector-valued RKHS and their associated multi-task kernels offer more natural and more powerful specialized procedures for addressing this issue.

In this work we adopt a multi-task kernel approach to take into account within- and between-view information. Multi-task kernels have been introduced in the machine learning community by Micchelli and Pontil in order to learn simultaneously multiple tasks (Evgeniou et al., 2005) and have recently been used successfully in functional regression (Kadri et al., 2011) and structured output prediction (Brouard et al., 2011; Kadri et al., 2013b), but to our knowledge have never been applied to the problem of multi-view learning until recently (Minh et al., 2013). We recall that the idea here is to consider the problem of learning from multiple views as a special case of multi-task learning where all tasks share the same label, and to learn the function $\mathbf{f}$ in a vector-valued RKHS. To this end, output values (labels) y_i are considered to be vectors in $\mathbb{R}^m$, where m is the number of views, instead of real variables as usual. The simplest representation would therefore consist of m repetitions of the real-valued labels, even if it might be possible to design more sophisticated representations to encode specific prior knowledge about the task into the model. By doing this, one can make use of multi-task kernels, which in this case are matrix-valued functions from $\mathcal{X} \times \mathcal{X} \to \mathbb{R}^{m\times m}$ (since $\mathcal{Y} = \mathbb{R}^m$), to exhibit the interesting multimodal structure of the data represented by different features. In the following we show how to construct such kernels, but before this we recall that the solution to the multi-view learning problem (3) in a vector-valued RKHS is given by the representer theorem, and has the following form

$$\mathbf{f}(\cdot) = \sum_{i\in\mathbb{N}} K(\cdot, x_i)c_i, \tag{4}$$

where $c_i \in \mathbb{R}^m$. The learning algorithm will then be based on computing and manipulating the kernel matrix associated with the matrix-valued function $K(\cdot,\cdot)$. In this setting, the kernel matrix is a block matrix of size $nm \times nm$, where each block of size $n \times n$ at the s-th row and the t-th column, $(s,t) \in (\mathbb{N}_m)^2$, is devoted to take into account information between views s and t. This is described in more details in the next subsection.

3.2. Multi-task kernels for multimodal data

In the multi-view learning setting and in order to encode within- and between-view information, we strongly prefer to view the multi-task kernel as a scalar-valued function rather than viewing it as a matrix-valued function. The multi-task kernel $K(\cdot,\cdot)$, which takes two examples x and x' and outputs a matrix of dimension $m \times m$, with m the number of views, can be reformulated using a scalar-valued function (kernel) k on two pairs of input/view (Wahba, 1992; Evgeniou et al., 2005) as follows

$$k(x,t,x',s) = \big(K(x,x')\big)_{ts},$$

where $(x,x') \in (\mathcal{X})^2$, $(s,t) \in (\mathbb{N}_m)^2$, and $K(x,x') \in \mathbb{R}^{m\times m}$. In this way, it is possible to construct the block kernel matrix $\mathbf{K} = \big(K(x_i,x_j)\big)_{(i,j)\in(\mathbb{N}_n)^2} \in \mathbb{R}^{nm\times nm}$ from blocks of size $n \times n$ over input features of each view (or source) instead of blocks of size $m \times m$ over outputs. Since the tasks share the same labels, this is more convenient in order to extract the inherent structure that is informative on the multimodal data.

Multi-task kernel based on cross-covariance operator. In order to take into account within- and between-view information, we consider multi-task kernels built from cross-covariance operators on RKHS. Covariance operators on RKHS have recently received considerable attention. For example, these operators that provide the simplest measure of dependency have been successfully applied to the problem of dimensionality reduction (Fukumizu et al., 2004) and played an important role in dealing with a number of statistical test problems (Gretton et al., 2005). More recently, covariance-based operator-valued kernels have been introduced in the context of structured output learning to capture the dependencies between the outputs as well as between the input and output variables (Kadri et al., 2013b). Here we expand the applicability of these kernels to multi-view learning setting. We propose to construct a multi-task kernel that can operate over multiple modalities as the following

$$k(x_i,t,x_j,s) = \big\langle k_t(x_i^{(t)},\cdot), C_{\mathcal{X}_t\mathcal{X}_s} k_s(x_j^{(s)},\cdot)\big\rangle, \tag{5}$$

where $C_{\mathcal{X}_t\mathcal{X}_s} : \mathcal{H}_s \to \mathcal{H}_t$ is the cross-covariance operator between the RKHSs $\mathcal{H}_s$ and $\mathcal{H}_t$. To compute the kernel k, we have to estimate the cross-covariance operator with given data. The empirical cross-covariance operator $\hat{C}^{(n)}_{\mathcal{X}_t\mathcal{X}_s}$ is given by

$$\hat{C}^{(n)}_{\mathcal{X}_t\mathcal{X}_s} = \frac{1}{n}\sum_{i=1}^{n} k_t(x_i^{(t)},\cdot) \otimes k_s(x_i^{(s)},\cdot),$$

where $\otimes$ is the tensor product defined by $(f \otimes g)h = \langle g, h \rangle f$, for all $f \in \mathcal{X}_t$ and $g, h \in \mathcal{X}_s$. Thus, by restricting the cross-covariance operator $C_{\mathcal{X}_t\mathcal{X}_s}$ to the n-dimensional subspaces spanned by $\{k_t(x_i^{(t)}, \cdot)\}_{i=1}^n$ and $\{k_s(x_i^{(s)}, \cdot)\}_{i=1}^n$, we can estimate the operator by

$$\hat{C}_{\mathcal{X}_t\mathcal{X}_s}^{(n)} = \mathbf{k}_t \mathbf{k}_s,$$

where $\mathbf{k}_t$ and $\mathbf{k}_s$ are the kernel matrices associated with the scalar-valued kernels k_t and k_s, respectively (Fukumizu et al., 2004). The multi-task kernel thus built permits us to encode the interactions between views s and t, and the resulting block kernel matrix $\mathbf{K} = \big(k(x_i, t, x_j, s)\big)_{(i,j)\in(\mathbb{N}_n)^2}^{(s,t)\in(\mathbb{N}_m)^2}$ allows to incorporate all the available information regarding the m views into the model.

Special cases. The kernel matrix built from the above multi-task kernel is a full block matrix in that it consists of all the cross-covariance blocks from the views. One can alternatively consider only within-view information and would not necessarily need between-view dependencies. In this case, the kernel matrix is block diagonal and the associated multi-task kernel is

$$k(x_i, t, x_j, s) = \delta_{ts}\big\langle k_t(x_i^{(t)}, \cdot), C_{\mathcal{X}_t\mathcal{X}_s} k_s(x_j^{(s)}, \cdot)\big\rangle,$$

where δ_{ts} is the Kronecker delta function: $\delta_{ts} = \{1 \text{ if } t = s,\ 0 \text{ otherwise}\}$. One particularly attractive instantiation of such a kernel is when the cross-covariance operator is replaced by the identity operator

$$k(x_i, t, x_j, s) = \delta_{ts}\big\langle k_t(x_i^{(t)}, \cdot), k_s(x_j^{(s)}, \cdot)\big\rangle. \tag{6}$$

The kernel matrix is then block diagonal and each block corresponds to the "scalar-valued" kernel matrix over one view (or source). In other works, applying the vector-valued RKHS framework using the multi-task kernel of Equation (6) is equivalent to using a basic "uniformly weighted" multiple kernel learning (MKL) approach (Bach et al., 2004).

4. Algorithm and Computational Aspects

We now discuss the multi-task kernel based algorithm used in this work to tackle the problem of learning from multiple views. In particular, based on the framework described above, we will concentrate on the vector-valued regularized least squares (RLS) algorithm, and we show how computational efficiency results well know in the scalar-valued case (Rifkin and Lippert, 2007) can be restated in the vector-valued setting.

4.1. Vector-valued RLS

Vector-valued RLS algorithm is based on the squared loss

$$V(y, \hat{y}) = \|y - \hat{y}\|_{\mathbb{R}^m}^2. \tag{7}$$

Substituting Equation (7) in the problem (3), the minimization problem associated with the multi-task formulation of supervised multi-view learning can be written as

$$\underset{\mathbf{f}(\cdot)\in\mathcal{H}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^{n}\|y_i - \mathbf{f}(x_i)\|^2 + \lambda\|\mathbf{f}\|_{\mathcal{H}}^2. \tag{8}$$

Using the representer theorem, the problem (8) becomes

$$\underset{\mathbf{c}\in\mathbb{R}^{nm}}{\arg\min}\ \frac{1}{n}\|\mathbf{y}-\mathbf{Kc}\|^2+\lambda\langle\mathbf{Kc},\mathbf{c}\rangle, \tag{9}$$

where $\mathbf{c}=(c_i)_{i\in\mathbb{N}_n}\in\mathbb{R}^{nm\times 1}$ ($c_i\in\mathbb{R}^m$), $\mathbf{y}=(y_i)_{i\in\mathbb{N}_n}\in\mathbb{R}^{nm\times 1}$, and $\mathbf{K}$ is the block kernel matrix such that $\mathbf{K}=\big(k(x_i,t,x_j,s)\big)_{i,j,s,t}\in\mathbb{R}^{nm\times nm}$, for all $(i,j)\in(\mathbb{N}_n)^2$ and $(s,t)\in(\mathbb{N}_m)^2$. Setting the derivative with respect to $\mathbf{c}$ to zero, we see that $\mathbf{c}$ must satisfy

$$(\mathbf{K}+\lambda I)\mathbf{c}=\mathbf{y}, \tag{10}$$

where I denotes an appropriately-sized identity matrix. The prediction of a new test point x is given by

$$\begin{aligned}\mathbf{f}(x)&=\sum_{i\in\mathbb{N}_n}K(x,x_i)c_i\\&=\mathbf{K}_x(\mathbf{K}+\lambda I)^{-1}\mathbf{y},\end{aligned}$$

where $\mathbf{K}_x$ is the $m\times nm$ matrix with $m\times m$ block entries $\big(K(x,x_i)\big)_{i\in\mathbb{N}_n}$. These results are very similar to those in the usual scalar-valued case. A major difference is the use of a block kernel matrix instead of a Gram matrix.

4.2. Leave-one-out procedure

In this subsection, we show how the classical result regarding computing in a closed form the leave-one-out solution in the case of regularized least squares (Wahba, 1990; Rifkin and Lippert, 2007) can be extended to the vector-valued setting. While the extension can be carried out in a direct way, it has, to the best of our knowledge, never been worked out before, and provides an adequate mechanism for finding a "good" value of the regularization parameter λ involved in the vector-valued RLS optimization problem (see Equation (8)).

We shall use the notation of (Rifkin and Lippert, 2007) . Let $\mathbf{f}_{S^i}$ be the solution of the vector-valued RLS function trained on S^i, where S^i is the data set with the i-th point removed from the training set $S=\{(x_j,y_j):j\in\mathbb{N}_n\}\subset(\mathcal{X}\times\mathcal{Y})^n$ ($\mathcal{Y}=\mathbb{R}^m$, where m is the number of views). We define $LOOV=\big(\mathbf{f}_{S^i}(x_i)\big)_{i\in\mathbb{N}_n}$ and $LOOE=\big(y_i-\mathbf{f}_{S^i}(x_i)\big)_{i\in\mathbb{N}_n}$ to be the vectors of leave-one-out values and errors over the training set. Let $\mathbf{y}^i=(y^i_j)_{j\in\mathbb{N}_n}$, where $y^i_j\in\mathbb{R}^m$, be the nm-dimensional vector defined by:

$$y^i_j=\begin{cases} y_j & j\neq i\\ \mathbf{f}_{S^i}(x_i) & j=i.\end{cases}$$

If we solve the vector-valued RLS problem (8) when replacing the outputs $\mathbf{y}=(y_j)_{j\in\mathbb{N}_n}$ by $\mathbf{y}^i$, the optimal solution will be $\mathbf{f}_{S^i}$. This can be seen by observing that, for any $f\in\mathcal{H}$,

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^{n}\|y^i_j-\mathbf{f}(x_i)\|^2+\lambda\|\mathbf{f}\|^2_{\mathcal{H}} &\geq \frac{1}{n}\sum_{j\neq i}\|y^i_j-\mathbf{f}(x_i)\|^2+\lambda\|\mathbf{f}\|^2_{\mathcal{H}}\\ &\geq \frac{1}{n}\sum_{j\neq i}\|y^i_j-\mathbf{f}_{S^i}(x_i)\|^2+\lambda\|\mathbf{f}_{S^i}\|^2_{\mathcal{H}}\\ &= \frac{1}{n}\sum_{j=1}^{n}\|y^i_j-\mathbf{f}_{S^i}(x_i)\|^2+\lambda\|\mathbf{f}_{S^i}\|^2_{\mathcal{H}}.\end{aligned}$$

The above inequality says that we can write the vector of expansion coefficients $\mathbf{c}^i$ of the predictor $\mathbf{f}_{S^i}$ given by the representer theorem as $\mathbf{c}^i = \mathbf{G}^{-1}\mathbf{y}^i$, where $\mathbf{G} = (\mathbf{K} + \lambda I)$ with $\mathbf{K} = \big(K(x_i, x_j)\big)_{(i,j)\in(\mathbb{N}_n)^2}$ the $nm \times nm$ block kernel matrix associated with the multi-task kernel $K(\cdot,\cdot)$ evaluated at the training points, and therefore

$$\mathbf{f}_{S^i}(x_i) = \mathbf{K}_{x_i}\mathbf{G}^{-1}\mathbf{y}^i,$$

where $\mathbf{K}_{x_i}$ is the $m \times nm$ matrix with $m \times m$ block entries $\big(K(x_i, x_j)\big)_{j\in\mathbb{N}_n}$. This allows us to express $\mathbf{f}_{S^i}(x_i)$ in terms of $\mathbf{f}_S(x_i)$

$$\begin{aligned}
\mathbf{f}_{S^i}(x_i) - \mathbf{f}_S(x_i) &= \mathbf{K}_{x_i}\mathbf{G}^{-1}\mathbf{y}^i - \mathbf{K}_{x_i}\mathbf{G}^{-1}\mathbf{y} \\
&= \sum_{j\in\mathbb{N}_n} [\mathbf{K}\mathbf{G}^{-1}]_{i,j}(y_j^i - y_j) \\
&= [\mathbf{K}\mathbf{G}^{-1}]_{i,i}(\mathbf{f}_{S^i}(x_i) - y_j),
\end{aligned}$$

which leads to

$$\mathbf{f}_{S^i}(x_i) = [I - (\mathbf{K}\mathbf{G}^{-1})_{i,i}]^{-1}\ [\mathbf{f}_S(x_i) - (\mathbf{K}\mathbf{G}^{-1})_{i,i}y_i].$$

Then,

$$\mathbf{f}_{S^i}(x_i) = [I - (\mathbf{K}\mathbf{G}^{-1})_{i,i}]^{-1}\ [\mathbf{K}_{x_i}\mathbf{G}^{-1}\mathbf{y} - (\mathbf{K}\mathbf{G}^{-1})_{i,i}y_i]. \tag{11}$$

From Equation (11), it is straightforward to see that

$$LOOV = [diag_b(I - \mathbf{K}\mathbf{G}^{-1})]^{-1}\ [\mathbf{K}\mathbf{G}^{-1}\mathbf{y} - diag_b(\mathbf{K}\mathbf{G}^{-1})\mathbf{y}],$$

where $diag_b$ denotes the block diagonal operator[3]. We thus obtain

$$\begin{aligned}
LOOE &= \mathbf{y} - LOOV \\
&= \mathbf{y} + [diag_b(I - \mathbf{K}\mathbf{G}^{-1})]^{-1}\ [diag_b(\mathbf{K}\mathbf{G}^{-1})\mathbf{y} - \mathbf{K}\mathbf{G}^{-1}\mathbf{y}] \\
&= [diag_b(I - \mathbf{K}\mathbf{G}^{-1})]^{-1}\ [diag_b(I - \mathbf{K}\mathbf{G}^{-1})\mathbf{y} + diag_b(\mathbf{K}\mathbf{G}^{-1})\mathbf{y} - \mathbf{K}\mathbf{G}^{-1}\mathbf{y}] \\
&= [diag_b(I - \mathbf{K}\mathbf{G}^{-1})]^{-1}\ [\mathbf{y} - \mathbf{K}\mathbf{G}^{-1}\mathbf{y}].
\end{aligned} \tag{12}$$

This expression can be simplified in a way that leads to better computational properties. Let D and Q be the square matrices such that $\mathbf{K} = QDQ^\top$. D and Q are generated by matrix diagonalization. We can note that

$$\begin{aligned}
\mathbf{K}\mathbf{G}^{-1} &= QDQ^\top Q(D + \lambda I)^{-1}Q^\top \\
&= QD(D + \lambda I)^{-1}Q^\top \\
&= Q(D + \lambda I - \lambda I)(D + \lambda I)^{-1}Q^\top \\
&= I - \lambda\mathbf{G}^{-1}.
\end{aligned} \tag{13}$$

Substituting Equation (13) in Equation (12), we obtain

$$\begin{aligned}
LOOE &= [diag_b(I - (I - \lambda\mathbf{G}^{-1}))]^{-1}\ [\mathbf{y} - (I - \lambda\mathbf{G}^{-1})\mathbf{y}] \\
&= [diag_b(\mathbf{G}^{-1})]^{-1}\ \mathbf{G}^{-1}\mathbf{y} \\
&= [diag_b(\mathbf{G}^{-1})]^{-1}\ \mathbf{c}.
\end{aligned} \tag{14}$$

[3] Given a block matrix M of size $nm \times nm$, $diag_b(M)$ is the block diagonal matrix satisfying $diag_b(M)_{i,i} = M_{i,i}$, where $M_{i,i}$ is the block at the ith row and the ith column of size $m \times m$ and $i \in \mathbb{N}_n$.

It is interesting to note that the leave-one-out solution in the scalar-valued case can be recovered from Equation (14) when the number of views m is equal to 1.

4.3. Efficient computation

As we have seen in the previous subsection, the leave-one-out error associated with a vector-valued RLS algorithm can be computed in a closed form that provides a convenient way to choose a "good" value for the regularization parameter λ. However, this needs the computation of the solution of the vector-valued RLS algorithm for each value of λ. Selecting the value of λ using leave-one-out procedure can be then computationally demanding, since we have to perform a matrix inversion which costs $O(n^3m^3)$ for every λ. To address this issue, we show, as with the scalar-valued case, how the vector of coefficients $\mathbf{c}$ (Equation (10)), solution of the vector-valued RLS problem, can be efficiently calculated for different values of λ.

Let the eigendecomposition of $\mathbf{K}$ be $\mathbf{K} = QDQ^\top$, where D is diagonal with $D_{i,i} \geq 0$ and $QQ^\top = I$. Then, the solution of the vector-valued RLS $\mathbf{c}$ can be expressed as

$$\begin{aligned}\mathbf{c} &= (\mathbf{K} + \lambda I)^{-1}\mathbf{y} \\ &= (QDQ^\top + \lambda I)^{-1}\mathbf{y} \\ &= \big(Q(D + \lambda I)Q^\top\big)^{-1}\mathbf{y} \\ &= Q(D + \lambda I)^{-1}Q^\top\mathbf{y}. \end{aligned} \tag{15}$$

The computational complexity of computing the eigendecomposition of $\mathbf{K}$ and calculating $Q^\top\mathbf{y}$ is $O(n^3m^3)$ and $O(n^2m^2)$, respectively. Once these two terms have been computed, they will be stored in the memory. Hence, since the inversion and the multiplication of the diagonal matrix $D + \lambda I$ with $Q^\top\mathbf{y}$ can be performed in $O(nm)$ time and the multiplication of the resulting matrix from left by Q costs $O(n^2m^2)$, the solution $\mathbf{c}$ can be calculated from Equation (15) for different values of the regularization parameter λ with the complexity of $O(n^2m^2)$.

Moreover, we can compute $[diag_b(\mathbf{G}^{-1})]^{-1}$ (see Equation (14)) in $O(n^2m^3)$ time instead of $O(n^3m^3)$. This can be seen from the following equation

$$(\mathbf{G}^{-1})_{(i,j)\in(\mathbb{N}_n)^2} = \big(Q(D + \lambda I)^{-1}Q^\top\big)_{(i,j)\in(\mathbb{N}_n)^2} = \sum_{l\in\mathbb{N}_n} Q_{i,l}(D_{l,l} + \lambda I)^{-1}Q_{j,l}, \tag{16}$$

where $M_{(i,j)\in(\mathbb{N}_n)^2}$denotes the (i,j)-th $m\times m$ block of the block matrix M of size $nm\times nm$. $(\mathbf{G}^{-1})_{(i,j)\in(\mathbb{N}_n)^2}$ can be calculated from Equation (16) in $O(nm^3)$ time. The computation of $[diag_b(\mathbf{G}^{-1})]^{-1}$ can then be performed in $O(n^2m^3)$, since the inverse of a block diagonal matrix is also a block diagonal matrix and can be found by inverting individually the blocks. Therefore, as the computational cost of the vector-valued RLS algorithm is $O(n^3m3)$, we can compute the solution coefficients $\mathbf{c}$ and the leave-one-out error $LOOE$ over a fine grid of λ without additional cost if we perform an eigendecomposition of the block kernel matrix $\mathbf{K}$.

4.4. Speeding-up training : sparse vector-valued RLS

Obviously, the computational complexity of the vector-valued RLS algorithm is $O(n^3m^3)$, and the training can be too costly with large numbers of training examples and/or views.

This complexity can be reduced via the Kronecker tensor product (Minh et al., 2013) ; however this is restricted to the particular case of separable multi-task kernels (or block diagonal kernel matrix). Here we are interested in the general case of any multi-task kernel, and consider a sparse approximation of vector-valued RLS in which only a part of the training instances has a non-zero coefficient in Equation (2) (Rifkin et al., 2003; Pahikkala et al., 2012).

Let $W \subseteq \mathbb{N}_n$ be a subset of $\mathbb{N}_n$ and $w = |W|$. By $M_w \in \mathbb{R}^{wm \times nm}$ we denote the block submatrix of $M \in \mathbb{R}^{nm \times nm}$ that contains only the block rows indexed by W. $M_{ww} \in \mathbb{R}^{wm \times wm}$ denotes a block submatrix of M having block rows and columns indexed by W. Following (Rifkin et al., 2003), we consider instead of Equation (2) a solution allowing only the training points indexed by W to have non-zero coefficients. More formally, we consider

$$\mathbf{f}(\cdot) = \sum_{i \in W} K(\cdot, x_i) c_i,$$

where the set W is selected in advance. In this case, the vector of expansion coefficients $\mathbf{c} = (c_i)_{i \in W}$, where $c_i \in \mathbb{R}^m$, is a wm-dimensional vector, and the minimization problem (9) associated with the vector-valued RLS algorithm becomes

$$\arg\min_{\mathbf{c} \in \mathbb{R}^{wm}} \frac{1}{n} \|\mathbf{y} - \mathbf{K}_w^\top \mathbf{c}\|^2 + \lambda \langle \mathbf{K}_{ww} \mathbf{c}, \mathbf{c} \rangle. \tag{17}$$

By setting the derivative of problem (17) with respect to $\mathbf{c}$ to zero, we obtain

$$(\mathbf{K}_w \mathbf{K}_w^\top + \lambda \mathbf{K}_{ww}) \mathbf{c} = \mathbf{K}_w \mathbf{y}. \tag{18}$$

Equation (18) shows that when we only allow a subset of size w points to have non-zero coefficients in the expansion, we can solve a wm by wm system of equations rather than an nm by nm system. Moreover, as in the scalar-valued case (Pahikkala et al., 2012), we can see that the matrices $\mathbf{K}_{ww}$ and $\mathbf{K}_w \mathbf{K}_w^\top = (\mathbf{K}\mathbf{K})_{ww}$ are principal sub matrices of the positive definite block kernel matrices $\mathbf{K}$ and $\mathbf{K}\mathbf{K}$, respectively. Then the matrix $\mathbf{T} = (\mathbf{K}_w \mathbf{K}_w^\top + \lambda \mathbf{K}_{ww})$ is also positive definite and invertible. The solution of the vector-valued RLS optimization is then given by

$$\mathbf{c} = \mathbf{T}^{-1} \mathbf{K}_w \mathbf{y}. \tag{19}$$

Hence, the matrix inversion involved in the vector-valued RLS algorithm can be performed by this procedure in $O(w^3 m^3)$ time (Equation (19)) instead of $O(n^3 m^3)$ (Equation (10)).

5. Experiments

In order to get a first experimental feedback of the usefulness of the cross-covariance operator within the vector-valued RLS algorithm, we performed experiments on the dataset `Animals With Attributes` (AwA) (Lampert et al., 2009), which features six views describing images from 50 concepts (i.e. labels) in a multiclass supervised setting. Each view is described through hundreds of float-typed attributes. We compare our algorithm with: (1) a "uniformly weighted" MKL with a least squares loss, which can be seen as a late fusion algorithm and can be derived from our framework, and (2) an early fusion SVM algorithm.

5.1. Dataset Animals with Attributes (AwA)

We chose the dataset AwA because it comes with 6 views over $30,475$ images to be classified within 50 concepts in a multi-class setting (only one class labels each example). As pointed out below, these views depict various focus on the numerical representation of images; they may be either correlated, complementary, redundant, or even conflicting.

- cq: (global) color histogram ($1 \times 1 + 2 \times 2 + 4 \times 4$ spatial pyramid, 128 bins each, each histogram L1-normalized).
- lss: local self similarity ($2,000$ entry codebook, raw bag-of-visual-word counts).
- phog: histogram of oriented gradients ($1 \times 1 + 2 \times 2 + 4 \times 4$ spatial pyramid, 12 bins each, each histogram L1-normalized or all zero).
- rgsift: rgSIFT descriptors ($2,000$ entry codebook, bag-of-visual-word counts, L1-normalized).
- sift: SIFT descriptors ($2,000$ entry codebook, raw bag-of-visual-word counts).
- surf: SURF descriptors ($2,000$ entry codebook, raw bag-of-visual-word counts)

5.2. Protocol

Since AwA is a multi-class learning problem, we processed a one-vs-all learning and testing protocol. For each concept, a classifier is trained from the n examples of the concept and $2n$ randomly selected examples among other concepts (n varies from 46 to 584, depending on the concept ; see Table 1). Each experiment is performed once with a random training set of size $3n$ (n is the number of examples of the current concept) and a random testing set of twice that size. In each experiment, the hyper-parameter λ is set using the LOOE method. All reported methods use a kernel: the χ^2 kernel is used everywhere, as usually done on histogram-based descriptors.

The accuracy of each algorithm is reported for the algorithms: MKL (uniformly weighted), MVK (our Multi-View learning method with cross-covariance multi-task Kernel), and an SVM processed on the Early fusion of views using SVM (E-SVM).

5.3. Results

Table 1 summarizes the performance results of the MKL, MVK, and E-SVM, on the dataset AwA[4]. It shows that our approach with multi-task cross-covariance kernel (MVK) performs better than the other approaches on many concepts, and in average. It is worthwhile noticing that when MVK is worse than MKL on a given concept, the difference may not be significative. Still, MVK performs better on the most difficult concepts if considering all the views (`mole` which has only 46 examples for learning, `Polar-bear` where only color-based views are informative, etc.). Besides, Having a large look on all concepts (*cf.* row ALL – mean accuracy), MVK performs better in accuracy than the other methods. These

[4] For more experimental results with the same data set, see (Kadri et al., 2013a).

class	n	MKL	MVK	E-SVM	class	n	MKL	MVK	E-SVM
antelope	273	**98.21**	95.34	90.89	hamster	421	62.03	**93.62**	83.57
grizzly+bear	440	74.77	**94.55**	86.76	squirrel	454	61.07	**95.07**	84.51
killer+whale	271	75.45	**92.92**	90.12	rhinoceros	350	**97.71**	97.12	87.80
beaver	93	**97.62**	91.48	83.17	rabbit	284	**98.13**	95.14	83.48
dalmatian	295	**98.07**	95.24	88.12	bat	201	50.03	**91.51**	82.46
persian+cat	357	**97.59**	92.91	86.91	giraffe	550	**96.39**	95.44	88.51
horse	411	84.60	**92.38**	87.66	wolf	273	**98.21**	93.33	86.34
german+shepherd	529	84.11	**95.24**	86.83	chihuahua	347	**97.73**	92.80	86.10
blue+whale	131	**95.11**	92.02	89.54	rat	152	**98.84**	93.37	85.73
siamese+cat	243	**98.41**	91.88	86.16	weasel	140	**99.07**	80.17	84.59
skunk	142	93.93	**96.56**	88.35	otter	363	**97.62**	94.46	83.99
mole	46	63.01	**94.97**	75.23	buffalo	280	**98.16**	94.14	88.02
tiger	273	**98.21**	95.36	87.58	zebra	385c	81.37	**98.32**	92.71
hippopotamus	362	**97.60**	93.04	87.08	giant+panda	471	59.25	**96.44**	89.73
leopard	304	**98.00**	94.78	89.11	deer	537	56.42	**90.64**	83.76
moose	361	83.51	**92.41**	85.14	bobcat	321	**97.90**	93.69	84.49
spider+monkey	176	91.28	**95.61**	92.09	pig	166	**98.90**	98.25	89.30
humpback+whale	358	**96.55**	94.53	90.46	lion	242	**98.41**	92.20	87.79
elephant	374	72.99	**94.36**	87.94	mouse	104	**99.10**	82.28	81.05
gorilla	402	**97.36**	95.15	89.40	polar+bear	408	64.01	**92.53**	87.28
ox	84	47.77	92.87	**93.33**	collie	584	49.67	94.06	**96.16**
fox	213	**98.60**	94.15	89.27	walrus	169	**97.14**	94.14	90.35
sheep	316	**97.93**	96.46	87.70	raccoon	317	**97.92**	94.89	87.57
seal	255	**97.01**	89.68	82.89	cow	329	**97.84**	91.11	86.07
chimpanzee	351	76.51	**89.06**	87.27	dolphin	341	67.18	**92.80**	89.16
ALL		86.686	**93.409**	87.190					

Table 1: Accuracy of unweighted MKL, our approach MVK based on cross-covariance multi-task kernel, and an early fusion with SVM (E-SVM), on the dataset AwA.

first experimental results acknowledge the relevance of our approach: information between-views is actually taken into account for a better accuracy. The overall performances of our approach are quite promising, and validate its pertinence.

6. Conclusion

The main point of this paper was to introduce and evaluate a method for learning from multiple views using a multi-task learning perspective. We have shown the ability of multi-task kernels in conjunction with cross-covariance operators on RKHS to capture the interesting multimodal structure of the data. Our first experimental results on a real-world dataset show that, in contrast to standard supervised multi-view learning methods, the vector-valued RLS algorithm with cross-covariance multi-task kernels allows to easily incorporate both within and between-view information. Future work will aim to extend the presented multi-view/multi-task ideas towards ranking problems.

Acknowledgments

We would like to thank Liva Ralaivola for helpful discussions. This research was funded by the Ministry of Higher Education and Research and the CNRS PEPS program in Biology-Mathematics-Informatics.

References

F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, pages 41–48, 2004.

M. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

A. B. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *ICML*, pages 137–144, 2006.

C. Brouard, F. D'Alche-Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *ICML*, pages 593–600, 2011.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. In Omnipress, editor, *Proceedings of the 21st Annual Conference on Learning Theory*, 2008.

N. Cesa-Bianchi, D. R. Hardoon, and G. Leen. Guest editorial: Learning from multiple sources. *Machine Learning*, 79(1-2):1–3, 2010.

M. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *UAI 2008*, 2008.

K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9:1757–1774, 2008.

T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: SVM-2K, theory and practice. In *NIPS*, pages 355–362, 2006.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory: 16th International Conference*. Springer-Verlag, 2005.

H. Kadri, A. Rabaoui, P. Preux, E. Duflos, and A. Rakotomamonjy. Functional regularized least squares classification with operator-valued kernels. In *ICML*, pages 993–1000, 2011.

H. Kadri, S. Ayache, C. Caponni, S. Koço, F. X. Dupé, and E. Morvant. The multi-task learning view of multimodal data. Technical report, CNRS, 2013a.

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In *ICML*, 2013b.

J. Kludas, E. Bruno, and S. Marchand-Maillet. Information fusion in multimedia information retrieval. In *Adaptive Multimedia Retrieval*, pages 147–159, 2007.

A. Kumar and H. Daumé III. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

A. Kumar, P. Rai, and H. Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, Granada, Spain, 2011.

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

Y. Luo, D. Tao, C. Xu, D. Li, and C. Xu. Vector-valued multi-view semi-supervised learning for multi-label image classification. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

C. Micchelli and M. Pontil. Kernels for multi–task learning.

C. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17: 177–204, 2005.

H. Q. Minh, L. Bazzani, and V. Murino. A unifying framework for vector-valued manifold regularization and multi-view learning. In *ICML*, 2013.

I. Muslea and C.A. Knoblock. Active learning with multiple views. *J. Artif. Intell. Res. (JAIR)*, 27:203–233, 2006.

T. Pahikkala, H. Suominen, and J. Boberg. Efficient cross-validation for kernelized least-squares regression with sparse basis expansions. *Machine Learning*, 87(3):381–407, June 2012.

M. Reisert and H. Burkhardt. Learning equivariant functions with matrix valued kernels. *Journal of Machine Learning Research*, 8:385–408, 2007.

R. Rifkin and R. A. Lippert. Notes on regularized least-squares. Technical Report MIT-CSAIL-TR-2007-025, Massachusetts Institute of Technology, 2007.

R. Rifkin, G. Yeo, and T. Poggio. Regularized least squares classification. *Advances in Learning Theory: Methods, Model and Applications NATO Science Series III: Computer and Systems Sciences*, 190:131–153, 2003.

B. Schölkopf and A. J. Smola. Learning with kernels: Support vector machines, regularization, optimization, and beyond. *MIT Press*, 2001.

L. Schwartz. Sous-espaces hilbertiens d'espaces vectoriels topologiques et noyaux associés (noyaux reproduisants). *Journal d'Analyse Mathématique*, 13:115–256, 1964.

V. Sindhwani and D. S. Rosenberg. An RKHS for multi-view learning and manifold co-regularization. In *ICML*, pages 976–983, 2008.

S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, UK, 2005.

G. Wahba. *Spline models for observational data.* Society for Industrial and Applied Mathematics(SIAM), 1990. ISBN 0-89871-244-0.

G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting*, volume 12 of *Proc. of the Santa Fe Institute*, pages 95–112. Addison Wesley, 1992.

W. Wang and Z. Zhou. On multi-view active learning and the combination with semi-supervised learning. In *ICML*, pages 1152–1159, 2008.