Kernel-Based Lifelong Multitask Multiview Learning

Rami Mowakeaa Seung-Jun Kim*
Dept. of Computer Science & Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD, USA
{ramo1,sjkim}@umbc.edu

Darren K. Emge
Combat Capabilities Development Command
Chemical Biological Center RDCB-DRC-P
Gunpowder, MD, USA
darren.k.emge.civ@mail.mil

***Abstract*—Lifelong learning capitalizes on the shared skill structure present in a stream of tasks that arrive over time to improve upon the performance of single-task learners. In contemporary lifelong learning applications, it is often the case that there are multiple sensing modalities or *views* associated with each task. A crucial aspect in lifelong multitask multiview learning is to capture not only the shared structure among the tasks but also across views effectively. In this work, a nonparametric kernel-based learning framework is adopted to model even nonlinear shared structures in the tasks and views in a flexible and robust way. An efficient lifelong learning formulation is derived by judicious approximation of the per-task learning objectives, based on which the shared skill libraries can be updated online in function space. Numerical tests verify the efficacy of the proposed approach.**

I. Introduction

Multitask learning exploits the shared structure among related tasks to learn classifiers that can improve upon those that are obtained from independent single-task learning [1]. In a lifelong learning scenario, the tasks are revealed sequentially over time, and the shared knowledge and skills for different tasks need to be continually transferred from the past-trained tasks to new ones and vice versa. Lifelong learning has been applied to various problems including supervised learning [2]–[4] and reinforcement learning [5], [6].

Multitask multiview (MTMV) learning attempts to learn from related tasks, where the data from each task contain one or more sensing modalities or *views*. The crucial aspect of MTMV learning is to capture the dual heterogeneity structure that exists across tasks and views jointly to improve the prediction capability over the learners that observe the data from each of the tasks and views separately [7]. MTMV learning finds applications in various areas ranging from data mining [7], classification [8], to medicine [9].

A bipartite graph was used in [7] to capture the connections between tasks and views, and an objective was designed to increase the consistency among them. A full-order tensor was used in a multilinear formulation to model task-view interactions [10]. A shared latent representation was employed with consistency imposed upon per-view libraries over labeled as well as unlabeled samples in [11]. A latent space model was adopted and the consistency across views and the task-specificity of views were captured in the latent codes [12]. These works, however, postulated rather simple shared structure with parametric classes of functions.

Our goal is to develop a flexible lifelong MTMV learning algorithm by adopting a nonlinear nonparametric shared structure model in the reproducing kernel Hilbert space (RKHS). The lifelong learning strategy is derived based on judicious approximation of per-task learning objectives, and the update of the shared skill libraries for different views is done in the function space [4], [6], [13]. The computational complexity of the kernel method is also curbed through an online sparsification method.

The rest of this paper is organized as follows. In Sec. II, the formulations for single-task single-view (STSV), single-task multiview (STMV), and MTMV learning are presented. In Sec. III, our proposed method for kernel lifelong MTMV learning is derived. The results from numerical experiment validate our proposed method in Sec. IV. Conclusions are offered in Sec. V.

Notations: $(\cdot)^\top$ represents the vector/matrix transpose, and $(\cdot)^\dagger$ the pseudoinverse. $\otimes$ denotes the kronecker product. $\mathbf{1}_{M\times N}$ denotes a matrix of size M-by-N with all entries equal to 1. $\mathbf{I}_M$ represents the M-by-M identity matrix. $\mathrm{diag}\{\cdot\}$ represents a diagonal matrix whose diagonal entries are equal to the elements listed in $\{\cdot\}$, and $\mathrm{bdiag}\{\cdot\}$ denotes a block diagonal matrix with the elements in $\{\cdot\}$ used as the diagonal blocks.

II. Problem Formulation

A. STSV Learning

Given a set of N samples $\mathbf{X}^{(1)} := [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_N^{(1)}] \in \mathbb{R}^{d_1\times N}$ and the corresponding labels $\mathbf{y} := [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$, where $^{(1)}$ signifies the single view, STSV learning entails estimation of a function $f^{(1)} : \mathbb{R}^{d_1} \to \mathbb{R}$ such that $f^{(1)}(\mathbf{x}_n^{(1)}) \approx y_n$, for $n = 1, \ldots, N$. We consider a space of functions, which is a RKHS $\mathcal{H}^{(1)}$ defined by a kernel function $\kappa^{(1)}(\mathbf{x}, \mathbf{x}') = \langle \phi^{(1)}(\mathbf{x}), \phi^{(1)}(\mathbf{x}')\rangle$, where $\phi^{(1)}(\cdot)$ is a nonlinear feature map, $\langle\cdot,\cdot\rangle$ the inner product, and the norm is defined as $\|f\|_{\mathcal{H}^{(1)}} := \sqrt{\langle f, f\rangle}$.

This work was supported in part by the MSI STEM Research & Development Consortium (MSRDC)/U.S. Army under grant W911SR-14-2-0001, and by the National Science Foundation under grant 1631838. *Corresponding author.

B. STMV Learning

In multiview learning, the samples are obtained from different sources termed *views*. Define $\mathbf{X} = \left[\mathbf{X}^{(1)\top}, \ldots, \mathbf{X}^{(V)\top}\right]^\top$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times N}$ are the features from view v, d_v is the number of features for the v-th view, and $\sum_{v=1}^{V} d_v = d$. As in the single-view case, the goal is to find a function f that maps the features $\mathbf{x}_n := [\mathbf{x}_n^{(1)\top}, \ldots, \mathbf{x}_n^{(V)\top}]^\top$ to the label y_n. To accommodate different views, let us estimate a collection of functions $\mathbf{f} := [f^{(1)}, \ldots, f^{(V)}]^\top$, where each $f^{(v)} \in \mathcal{H}^{(v)}$ is associated with nonlinear feature map $\phi^{(v)}(\cdot)$. The desired function f is obtained as

$$f(\mathbf{x}_n) := \sum_{v=1}^{V} \langle f^{(v)}, \phi^{(v)}(\mathbf{x}_n^{(v)}) \rangle. \tag{1}$$

Then, a STMV learning problem can be formulated as

$$\min_{\mathbf{f}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(\sum_{v=1}^{V} \langle f^{(v)}, \phi^{(v)}(\mathbf{x}_n^{(v)}) \rangle, y_n \right) + \gamma \sum_{v=1}^{V} \|f^{(v)}\|^2_{\mathcal{H}^{(v)}} \tag{2}$$

where $\mathcal{L}$ is a twice-differentiable loss function, and $\gamma \geq 0$ is a regularization parameter that balances the complexity of the estimated functions and the fidelity to the training data. Given a collection of samples, the Representer Theorem [14] reduces specifying each function $f^{(v)}$ to finding a suitable linear combination of the lifted features $\Phi^{(v)}(\mathbf{X}^{(v)}) := [\phi(\mathbf{x}_1^{(v)}), \ldots, \phi(\mathbf{x}_N^{(v)})]$, yielding $f^{(v)} = \Phi^{(v)}(\mathbf{X}^{(v)})\mathbf{w}^{(v)}$.

C. MTMV Learning

In MTMV learning, one is presented with a collection of tasks $\{\mathcal{Z}_t := (\mathbf{X}_t, \mathbf{y}_t)\}$, where $\mathcal{Z}_t$ represents the multiview data $\mathbf{X}_t := [\mathbf{X}_t^{(1)\top}, \ldots, \mathbf{X}_t^{(V)\top}]^\top \in \mathbb{R}^{d \times N_t}$ and labels $\mathbf{y}_t := [y_{t,1}, \ldots, y_{t,N_t}]^\top$ for task $t = 1, \ldots, T$. For each task t, the purpose is to estimate $\mathbf{f}_t = [f_t^{(1)}, \ldots, f_t^{(V)}]^\top$ as in Sec. II-B. However, rather than solving a STMV problem for each task independently, shared structure across tasks is exploited.

Specifically, let us hypothesize that the functions corresponding to a view approximately adhere to a sparse latent model. That is, it is postulated that for view $v \in \{1, \ldots, V\}$

$$f_t^{(v)} \approx \mathbf{L}^{(v)} \mathbf{s}_t^{(v)}, \ t = 1, \ldots, T \tag{3}$$

where $\mathbf{L}^{(v)} := \left[\ell_1^{(v)}, \ldots, \ell_K^{(v)}\right]$ is a view-specific shared library with "skills" $\ell_k^{(v)} \in \mathcal{H}^{(v)}$ and $\mathbf{s}_t^{(v)} \in \mathbb{R}^K$ is the sparse code for the t-th task that linearly combines a small number of skills from library $\mathbf{L}^{(v)}$.

In order to enforce consistency across different views, first aggregate the latent codes of task t across views into a matrix $\mathbf{S}_t := [\mathbf{s}_t^{(1)}, \ldots, \mathbf{s}_t^{(V)}] \in \mathbb{R}^{K \times V}$. Then, it is postulated that $\mathbf{S}_t$ can be decomposed as $\mathbf{S}_t = \mathbf{P}_t + \mathbf{Q}_t$, where $\mathbf{P}_t$ is row-sparse and $\mathbf{Q}_t$ is column-sparse [12], [15]. Promoting row-sparsity in $\mathbf{P}_t$ encourages task t to utilize a few common skills across views, thereby effecting skill consistency. On the other hand, column-sparse $\mathbf{Q}_t$ allows task t to employ any skills for select few views that do not conform to the aforementioned skill consistency, thus providing robustness to the model.

Putting these elements together, our MTMV problem can be formulated as

$$\min_{\{\mathbf{L}^{(v)}\}} \frac{1}{T} \sum_{t=1}^{T} \min_{\mathbf{P}_t, \mathbf{Q}_t} \left\{ \frac{1}{N_t} \sum_{n=1}^{N_t} \mathcal{L}\Bigg(\sum_{v=1}^{V} \left\langle \mathbf{L}^{(v)} \mathbf{s}_t^{(v)}, \phi^{(v)}(\mathbf{x}_{t,n}^{(v)}) \right\rangle, y_{t,n} \Bigg) + \mu_1 \|\mathbf{P}_t\|_{2,1} + \mu_2 \left\|\mathbf{Q}_t^\top\right\|_{2,1} \right\} + \sum_{v=1}^{V} \lambda_v \|\mathbf{L}^{(v)}\|^2_{\mathcal{H}^{(v)}}$$
$$\text{s.t. } \mathbf{S}_t = \mathbf{P}_t + \mathbf{Q}_t, \ t = 1 \ldots, T \tag{4}$$

where the $\ell_{2,1}$-norm of $\mathbf{P} := [\mathbf{p}_1, \ldots, \mathbf{p}_R]^\top$ is defined as $\|\mathbf{P}\|_{2,1} := \sum_{r=1}^{R} \|\mathbf{p}_r\|_2$, μ_1, μ_2, and $\{\lambda_v\}$ are preselected nonnegative parameters, and $\|\mathbf{L}^{(v)}\|^2_{\mathcal{H}^{(v)}} := \sum_{k=1}^{K} \|\ell_k^{(v)}\|^2_{\mathcal{H}^{(v)}}$.

III. Lifelong MTMV Learning

A. Lifelong MTMV Formulation

Solving the batch formulation in (4) becomes challenging when the tasks keep arriving continually. A viable alternative is to perform lifelong learning, where the library is updated in an online fashion.

First, it is noted that (4) involves the data from all tasks and views, incurring high storage and computational burdens. A useful idea is to eliminate the dependency on the past tasks' samples by employing an appropriate local approximation of the STMV learning objective.

Denote the objective function of (2) for task t as $\mathcal{J}_t(\mathbf{f})$. Then, a quadratic approximation of $\mathcal{J}_t(\mathbf{f})$ around its minimum $\mathbf{f}_t^*$ can be adopted [4]–[6]. Since $\mathcal{L}(\cdot)$ is twice-differentiable, the gradient of $\mathcal{J}_t$ is given by $\nabla_{\mathbf{f}} \mathcal{J}_t(\mathbf{f}) = \left[\nabla_{f^{(1)}} \mathcal{J}_t(\mathbf{f}), \ldots, \nabla_{f^{(V)}} \mathcal{J}_t(\mathbf{f})\right]^\top$ where

$$\nabla_{f^{(v)}} \mathcal{J}_t(\mathbf{f}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \mathcal{L}'_{t,n} \phi^{(v)}(\mathbf{x}_{t,n}^{(v)}) \tag{5}$$

and $\mathcal{L}'_{t,n} := \partial \mathcal{L}(\hat{y}_{t,n}, y_{t,n}) / \partial \hat{y}_{t,n}$. Similarly, the Hessian of $\mathcal{J}_t$ is given as

$$\mathbf{H}_t := \begin{bmatrix} \mathbf{H}_t^{(1,1)} & \ldots & \mathbf{H}_t^{(1,V)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_t^{(V,1)} & \ldots & \mathbf{H}_t^{(V,V)} \end{bmatrix} \tag{6}$$

where for $u, v \in \{1, \ldots, V\}$

$$\mathbf{H}_t^{(u,v)} := \nabla^2_{f^{(u)}, f^{(v)}} \mathcal{J}_t(\mathbf{f}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \mathcal{L}''_{t,n} \phi(\mathbf{x}_{t,n}^{(u)}) \phi^\top(\mathbf{x}_{t,n}^{(v)}) \tag{7}$$

and $\mathcal{L}''_{t,n} := \partial^2 \mathcal{L}(\hat{y}_{t,n}, y_{t,n}) / \partial \hat{y}^2_{t,n}$. Define a block diagonal matrix consisting of the view-specific libraries as

$$\mathbf{L} := \text{bdiag}\{\mathbf{L}^{(1)}, \ldots, \mathbf{L}^{(V)}\} \tag{8}$$

and the task t's code vector $\mathbf{s}_t := [\mathbf{s}_t^{(1)\top}, \ldots, \mathbf{s}_t^{(V)\top}]^\top$. Then, by substituting the loss function in (4) with the second-order

Taylor series expansion of $\mathcal{J}_t$, and dropping the constant terms, the problem can be re-written as

$$\min_{\{\mathbf{L}^{(v)}\}} \frac{1}{T}\sum_{t=1}^{T} \min_{\mathbf{P}_t,\mathbf{Q}_t} \Big\{ \frac{1}{2}\|\mathbf{L}\mathbf{s}_t - \mathbf{f}_t^*\|_{\mathbf{H}_t}^2 + \mu_1\|\mathbf{P}_t\|_{2,1} + \mu_2\|\mathbf{Q}_t^\top\|_{2,1}\Big\} + \sum_{v=1}^{V}\lambda_v\|\mathbf{L}^{(v)}\|_{\mathcal{H}^{(v)}}^2 \quad \text{s.t. } \mathbf{S}_t = \mathbf{P}_t + \mathbf{Q}_t,\ t=1,\ldots,T \tag{9}$$

where $\|\mathbf{v}\|_{\mathbf{H}}^2 := \mathbf{v}^\top\mathbf{H}\mathbf{v}$. Problem (9) is amenable to online learning by optimizing the objective over $\mathbf{P}_t$ and $\mathbf{Q}_t$ upon arrival of task t (without updating the past $\mathbf{P}_\tau$'s and $\mathbf{Q}_\tau$'s for $\tau < t$), followed by updating $\{\mathbf{L}^{(v)}\}$, as detailed next [13].

B. Online Library Update in Function Space

In lifelong learning, the goal is to find the solution to (9) in an online fashion as the number of tasks grows—possibly indefinitely. To that end, it is first noted that upon defining the instantaneous cost

$$g\left(\mathbf{L};\mathbf{f}_t^*,\mathbf{H}_t\right) := \min_{\mathbf{P}_t,\mathbf{Q}_t}\Big\{\frac{1}{2}\|\mathbf{L}\mathbf{s}_t-\mathbf{f}_t^*\|_{\mathbf{H}_t}^2 + \mu_1\|\mathbf{P}_t\|_{2,1} + \mu_2\|\mathbf{Q}_t^\top\|_{2,1}\Big\} \tag{10}$$

the objective of (9) tends to

$$\mathbb{E}\left\{g(\mathbf{L};\mathbf{f}_t^*,\mathbf{H}_t)\right\} + \sum_{v=1}^{V}\lambda_v\|\mathbf{L}^{(v)}\|_{\mathcal{H}^{(v)}}^2 \tag{11}$$

as the number T of tasks increases due to the Law of Large Numbers, where the expectation is taken with respect to $\mathbf{f}_t^*$ and $\mathbf{H}_t$.

Suppose that at the current iteration t, the aggregate library is denoted by $\mathbf{L}(t-1)$ and a new task is presented. First, the single-task optimum and the associated Hessian are computed from the incoming task by solving (2). Next, $\mathbf{P}_t$ and $\mathbf{Q}_t$ are computed by solving (10). Then, using stochastic gradient descent (SGD) in function space, the aggregate library is updated as

$$\mathbf{L}(t) = \mathbf{L}(t-1) - \eta\nabla_{\mathbf{L}}\left\{g\left(\mathbf{L}(t-1);\mathbf{f}_t^*,\mathbf{H}_t\right) + \sum_{v=1}^{V}\lambda_v\|\mathbf{L}^{(v)}\|_{\mathcal{H}^{(v)}}^2\right\} \tag{12}$$

where $\eta > 0$ is a small step size and

$$\nabla_{\mathbf{L}}\left\{g\left(\mathbf{L}(t-1);\mathbf{f}_t^*,\mathbf{H}_t\right)\right\} = \mathbf{H}_t\left(\mathbf{L}(t-1)\mathbf{s}_t - \mathbf{f}_t^*\right)\mathbf{s}_t^\top. \tag{13}$$

Performing the library update in function space as in (12) is made tractable through the dual updates in finite dimensions. To see this, note that the Representer Theorem allows the description of $\mathbf{L}$ as a linear combination of the lifted features of all observed samples. That is,

$$\mathbf{L} = \begin{bmatrix}\Phi^{(1)}(\mathbf{D}^{(1)}) & & \\ & \ddots & \\ & & \Phi^{(V)}(\mathbf{D}^{(V)})\end{bmatrix}\begin{bmatrix}\mathbf{A}^{(1)} & & \\ & \ddots & \\ & & \mathbf{A}^{(V)}\end{bmatrix}. \tag{14}$$

For example, at iteration $t-1$, $\mathbf{L}(t-1)$ can be represented using the pool of samples $\mathbf{D}^{(v)}(t-1) = [\mathbf{X}_1^{(v)},\ldots,\mathbf{X}_{t-1}^{(v)}]$ and coefficients $\mathbf{A}^{(v)}(t-1) \in \mathbb{R}^{(\sum_{\tau=1}^{t-1}N_\tau)\times K}$, $v=1,\ldots,V$. At iteration t, when task t arrives, the new set of samples $\mathbf{X}_t^{(v)}$ are appended to the pool as

$$\mathbf{D}^{(v)}(t) = [\mathbf{D}^{(v)}(t-1), \mathbf{X}_t^{(v)}],\ v=1,\ldots,V. \tag{15}$$

The coefficient matrices $\{\mathbf{A}^{(v)}(t)\}$ correspondingly grow in size by N_t rows.

To derive the update equations in finite dimensions, let us define a block diagonal matrix

$$\mathbf{\Phi}(t) := \mathrm{bdiag}\left\{\Phi^{(v)}\left([\mathbf{D}^{(v)}(t-1),\mathbf{X}_t^{(v)}]\right)\right\}_{v=1}^{V}. \tag{16}$$

The aggregate library at time t can then be represented as $\mathbf{L}(t) = \mathbf{\Phi}(t)\mathbf{A}(t)$ with $\mathbf{A}(t) := \mathrm{bdiag}\{\mathbf{A}^{(1)}(t),\ldots,\mathbf{A}^{(V)}(t)\}$. At time $t-1$, it can be expressed as

$$\mathbf{L}(t-1) = \mathbf{\Phi}(t)\cdot\mathrm{bdiag}\left\{\begin{bmatrix}\mathbf{A}^{(v)}(t-1)\\ \mathbf{0}\end{bmatrix}\right\}_{v=1}^{V}. \tag{17}$$

Note also that upon defining a $V(\sum_{\tau=1}^{t}N_\tau)\times VN_t$ matrix

$$\boldsymbol{\mathcal{L}}''(t) := \mathbf{I}_V \otimes \begin{bmatrix}\mathbf{0}_{(\sum_{\tau=1}^{t-1}N_\tau)\times N_t}\\ \mathrm{diag}\{\mathcal{L}''_{t,1},\ldots,\mathcal{L}''_{t,N_t}\}\end{bmatrix} \tag{18}$$

the Hessian $\mathbf{H}_t$ can be expressed as

$$\mathbf{H}_t = \mathbf{\Phi}_{1:t}\boldsymbol{\mathcal{L}}''(t)\mathbf{1}_{V\times 1}\otimes[\Phi^{(1)}(\mathbf{X}_t^{(1)})^\top,\ldots,\Phi^{(V)}(\mathbf{X}_t^{(V)})^\top]. \tag{19}$$

By incorporating these definitions into (12) and matching the terms on both sides of the equation, one obtains

$$\mathbf{A}^{(v)}(t) = \begin{bmatrix}(1-2\lambda_v\eta)\mathbf{A}^{(v)}(t-1)\\ \tilde{\mathbf{A}}^{(v)}(t)\end{bmatrix},\ v=1,\ldots,V \tag{20}$$

where $\{\tilde{\mathbf{A}}^{(v)}(t)\}_{v=1}^{V}$ can be found by extracting the V diagonal blocks of size $N_t\times K$ from

$$-\frac{\eta}{N_t}\mathbf{I}_V\otimes\mathrm{diag}\{\mathcal{L}''_{t,1},\ldots,\mathcal{L}''_{t,N_t}\}\cdot[\mathbf{K}_1\mathbf{A}(t-1)\mathbf{s}_t + \mathbf{K}_2\mathbf{w}_t^*]\,\mathbf{s}_t^\top \in \mathbb{R}^{VN_t\times VK} \tag{21}$$

where $\mathbf{K}_{\mathbf{X},\mathbf{X}'}^{(v)} := \Phi^{(v)}(\mathbf{X})^\top\Phi^{(v)}(\mathbf{X}')$,

$$\mathbf{K}_1 := \mathbf{1}_{V\times 1}\otimes\left[\mathbf{K}^{(1)}_{\mathbf{X}_t^{(1)},\mathbf{D}^{(1)}(t-1)},\ldots,\mathbf{K}^{(V)}_{\mathbf{X}_t^{(V)},\mathbf{D}^{(V)}(t-1)}\right] \tag{22}$$

$$\mathbf{K}_2 := \mathbf{1}_{V\times 1}\otimes\left[\mathbf{K}^{(1)}_{\mathbf{X}_t^{(1)},\mathbf{X}_t^{(1)}},\ldots,\mathbf{K}^{(V)}_{\mathbf{X}_t^{(V)},\mathbf{X}_t^{(V)}}\right] \tag{23}$$

and $\mathbf{w}_t^*$ is the dual coefficient vector for $\mathbf{f}_t^*$, i.e., $\mathbf{w}_t^* = [\mathbf{w}_t^{(1)*\top},\ldots,\mathbf{w}_t^{(V)*\top}]^\top$ with $f_t^{(v)*} = \Phi^{(v)}(\mathbf{X}_t^{(v)})\mathbf{w}_t^{(v)*}$, $v=1,\ldots,V$.

TABLE I
KERNEL LIFELONG MTMV LEARNING ALGORITHM.

Input: Multiview data $\mathbf{X}_t$, labels $\mathbf{y}_t$, $t = 1, 2, \ldots$ and parameters $\{\lambda_v\}, \mu_1, \mu_2, \eta, \epsilon, K$
Output: Per-view libraries $\{\mathbf{L}^{(v)} = \Phi^{(v)}(\mathbf{D}^{(v)})\mathbf{A}^{(v)}\}$, and latent codes $\{\mathbf{P}_t\}$, $\{\mathbf{Q}_t\}$, $t = 1, 2, \ldots$
1: Initialize libraries $\{\mathbf{L}^{(v)}(0)\}$ randomly or with first K $\mathbf{f}_t^*$'s
2: For $t = 1, 2, \ldots$
3: Compute STMV optimum $\mathbf{f}_t^*$ by solving (2) in batch or online
4: Compute the corresponding Hessian $\mathbf{H}_t$ via (6)–(7)
5: Obtain $\mathbf{P}_t$ and $\mathbf{Q}_t$ by solving (10)
6: Obtain $\{\check{\mathbf{L}}^{(v)}(t)\}$ using (24) and (20)
7: Sparsify the pool elements to obtain $\{\mathbf{D}^{(v)}(t)\}$ and project the libraries via (26) to produce parsimonious libraries $\{\mathbf{L}^{(v)}(t)\}$
8: End For

C. Parsimonious Library Representation

As in any kernel-based learning methods, the solution obtained in Sec. III-B depends on all observed training samples. This is particularly acute in lifelong learning as the number of tasks can grow indefinitely, accumulating a prohibitive number of samples. To mitigate the ensuing computational and memory complexity, a method to prune the sample pool needs to be adopted. Inspired by [16], we propose to discard as many samples as possible from the pools $\{\mathbf{D}^{(v)}(t)\}$ at each iteration t as long as the libraries $\{\mathbf{L}^{(v)}(t)\}$ can be approximated within a specified tolerance in the Hilbert norm sense.

Starting from the pools $\{\mathbf{D}^{(v)}(t-1)\}$ from iteration $t-1$, upon arrival of task t, append the new set of samples $\{\mathbf{X}_t^{(v)}\}$ as in (15) but now denote the updated pool as $\check{\mathbf{D}}^{(v)}(t)$, i.e.,

$$\check{\mathbf{D}}^{(v)}(t) := [\mathbf{D}^{(v)}(t-1), \mathbf{X}_t^{(v)}], \ v = 1, \ldots, V. \tag{24}$$

By performing the update in (20), one can get the updated coefficient matrix, which we will denote as $\check{\mathbf{A}}^{(v)}(t)$ now. Define $\check{\mathbf{L}}^{(v)}(t) := \Phi^{(v)}(\check{\mathbf{D}}^{(v)}(t))\check{\mathbf{A}}^{(v)}(t)$ for $v = 1, \ldots, V$. Then, a greedy algorithm called the destructive kernel orthogonal matching pursuit (KOMP) algorithm is employed to discard one by one the samples (columns) in $\check{\mathbf{D}}^{(v)}(t)$ to find the smallest subset $\mathbf{D}^{(v)}(t)$ of $\check{\mathbf{D}}^{(v)}(t)$ that satisfies a specified error tolerance [16]. That is, the approximation error given by

$$\max_{v\in\{1,\ldots,V\}} \min_{\mathbf{L}^{(v)}\in \text{span}\{\Phi^{(v)}(\mathbf{D}^{(v)}(t))\}} \|\mathbf{L}^{(v)} - \check{\mathbf{L}}^{(v)}(t)\|^2_{\mathcal{H}^{(v)}} \tag{25}$$

is ensured not to exceed a pre-specified tolerance ϵ. The set of indices of the discarded samples is constrained to be identical across all views. The projection (minimization) in (25) can be computed through

$$\begin{aligned}
&\mathbf{A}^{(v)}(t) \\
&= \arg\min_{\mathbf{A}} \|\Phi^{(v)}(\mathbf{D}^{(v)}(t))\mathbf{A} - \Phi^{(v)}(\check{\mathbf{D}}^{(v)}(t))\check{\mathbf{A}}^{(v)}(t)\|^2_{\mathcal{H}^{(v)}} \\
&= \mathbf{K}^{(v)\dagger}_{\mathbf{D}^{(v)}(t),\mathbf{D}^{(v)}(t)} \mathbf{K}^{(v)}_{\mathbf{D}^{(v)}(t),\check{\mathbf{D}}^{(v)}(t)} \check{\mathbf{A}}^{(v)}(t)
\end{aligned} \tag{26}$$

from which one can get $\mathbf{L}^{(v)}(t) = \Phi^{(v)}(\mathbf{D}^{(v)}(t))\mathbf{A}^{(v)}(t)$ for $v = 1, \ldots, V$. The overall algorithm is described in Table I.

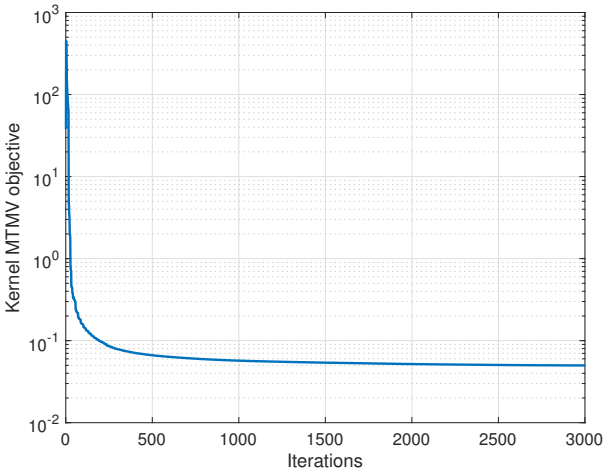

Fig. 1. Convergence of the proposed algorithm.

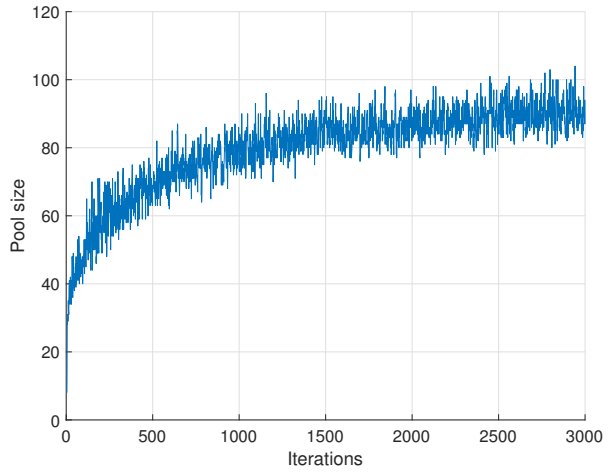

Fig. 2. Evolution of the pool size.

IV. NUMERICAL TESTS

To demonstrate the effectiveness of the proposed approach, regression tasks with $V = 5$ views were considered based on synthetic data sets. First, $K = 3$ task clusters with centroids $\{\mathbf{f}_{c,k} := [f_{c,k}^{(1)}, \ldots, f_{c,k}^{(5)}]^\top\}_{k=1}^3$ were generated through

$$f_{c,k}^{(v)} = \Phi^{(v)}(\mathbf{D}^{(v)})\mathbf{c}_k^{(v)}, \ k = 1, \ldots, 3, \ v = 1, \ldots, 5 \tag{27}$$

where the elements of $\mathbf{D}^{(v)} \in \mathbb{R}^{20\times 100}$ and $\mathbf{c}_k^{(v)} \in \mathbb{R}^{100}$ were randomly drawn from Gaussian distributions $\mathcal{N}(0, 4)$ and $\mathcal{N}(0, 1)$, respectively. Then, the ground truth mappings for $T = 15$ tasks $\{\mathbf{f}_t\}$ were generated via

$$f_t^{(v)} = f_{c,k_t}^{(v)} + \Phi^{(v)}(\mathbf{D}^{(v)})\mathbf{b}_t^{(v)}, \ t = 1, \ldots, 15, \ v = 1, \ldots, 5 \tag{28}$$

where k_t is a cluster index for task t chosen uniformly over $\{1, 2, 3\}$. The coefficients $\mathbf{b}_t^{(v)} \in \mathbb{R}^{100}$ is equal to $\mathbf{0}$ for all views except for one randomly chosen view $\bar{v}$, for which $\mathbf{b}_t^{(\bar{v})}$ has elements drawn from $\mathcal{N}(0, 1)$. The first term in (28)

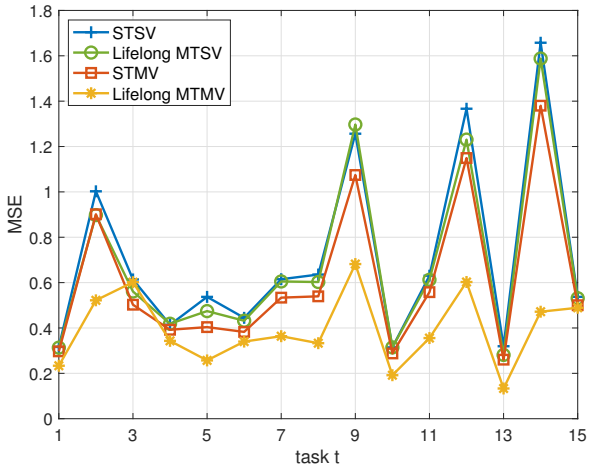

Fig. 3. Performance comparison.

TABLE II
MSEs AND NORMALIZED MSEs AVERAGED OVER TASKS.

	STSV	Lifelong MTSV	STMV	Lifelong MTMV
MSE	0.71	0.68	0.61	0.39
NMSE	1.07	1.02	0.92	0.59

captures a common skill across views, while the second term represents an aberration present in some random view. Then, 200 samples were generated for each task according to $\mathbf{x}_{t,n}^{(v)} \sim \mathcal{N}((2v-6)\mathbf{1}_{20\times 1}, 0.1\mathbf{I}_{20})$ and the corresponding ground truth labels were obtained by $y_{t,n} = \sum_{v=1}^{V} f_t^{(v)}(\mathbf{x}_{t,n}^{(v)}) + \mathcal{N}(0, 0.01)$. Finally, the data set was divided into 10% training, 45% validation, and 45% test samples.

In Fig. 1, the evolution of the kernel lifelong MTMV learning objective function in (4) is shown. It can be seen that the algorithm converges. Fig. 2 depicts the evolution of the number of the pool elements needed to represent each of the per-view libraries. Interestingly, although our algorithm is never privy to the particular set of samples used to generate the ground truth, the pool size tends to stabilize at around 90 elements—on par with the 100 used in the ground truth libraries.

In Fig. 3, the per-task mean-square error (MSE) performances of the proposed lifelong MTMV algorithm is shown. For comparison, the MSEs of STSV learning, lifelong MTSV learning, and STMV learning are also depicted. For single-view algorithms, the MSE is averaged over all views. It is observed that while lifelong MTSV learning improves upon the STSV learning slightly, much more improvement is achieved by STMV learning as the views are selected robustly. However, the best performance is attained through proposed lifelong MTMV learning, which is seen to perform markedly better than the others, particularly in the challenging tasks, by effectively learning from other tasks. Table II lists the MSEs and the normalized MSEs (NMSEs), normalized by the variance of the ground truth labels, averaged over tasks, achieved by different methods.

V. Conclusions

A kernel-based lifelong MTMV learning algorithm that can exploit the inherent shared structure across related tasks as well as multiple views has been proposed. A sparse latent model was adopted in function space to learn the skill library shared across tasks for each view. Skill consistency across views was robustly enforced through appropriate group sparsity constraints. The resulting batch MTMV problem was then approximately re-formulated to derive practicable online update rules suitable for the lifelong learning setting. The SGD-based library updates in function space were shown to be implementable in finite dimensions thanks to the Representer Theorem. The growing pool of samples needed to represent the solution was pruned parsimoniously. The numerical tests verified the effectiveness of our proposed approach.

References

[1] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[2] G. Pillonetto, F. Dinuzzo, and G. De Nicolao, "Bayesian online multitask learning of Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 193–205, 2010.

[3] P. Ruvolo and E. Eaton, "ELLA: An efficient lifelong learning algorithm," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, Jun. 2013, pp. 507–515.

[4] S.-J. Kim and R. Mowakeaa, "Kernel-based efficient lifelong learning algorithm," in *Proc. IEEE Data Sci. Workshop*, Minneapolis, MN, Jun. 2019.

[5] H. Ammar, E. Eaton, P. Ruvolo, and M. Taylor, "Online multi-task learning for policy gradient methods," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 1206–1214.

[6] R. Mowakeaa and S.-J. Kim, "Kernel-based lifelong policy gradient reinforcement learning," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Toronto, Canada, Jun. 2021.

[7] J. He and R. Lawrence, "A graph-based framework for multi-task multi-view learning," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, Jun. 2011, pp. 25–32.

[8] X. Jin, F. Zhuang, S. Wang, Q. He, and Z. Shi, "Shared structure learning for multiple tasks with multiple views," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Prague, Czech Republic, Sep. 2013, pp. 353–368.

[9] L. Nie, L. Zhang, L. Meng, X. Song, X. Chang, and X. Li, "Modeling disease progression via multisource multitask learning: A case study with Alzheimer's disease," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1508–1519, Jul. 2017.

[10] C.-T. Lu, L. He, W. Shao, B. Cao, and P. S. Yu, "Multilinear factorization machines for multi-task multi-view learning," in *Proc. of the 10th ACM Int. Conf. Web Search Data Mining*, Cambridge, UK, Feb. 2017, pp. 701–709.

[11] X. Li, S. Chandrasekaran, and J. Huan, "Lifelong multi-task multi-view learning using latent spaces," in *Proc. IEEE Int. Conf. Big Data*, Boston, MA, Dec. 2017, pp. 37–46.

[12] G. Sun, Y. Cong, J. Li, and Y. Fu, "Robust lifelong multi-task multi-view representation learning," in *Prof. IEEE Int. Conf. Big Knowl.*, Singapore, Nov. 2018, pp. 91–98.

[13] S.-J. Kim, "Online kernel dictionary learning," in *Proc. IEEE Global Conf. Signal and Info. Process.*, Orlando, FL, Dec. 2015, pp. 103–107.

[14] B. Schölkopf, A. Smola, and F. Bach, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.

[15] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. of the 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Beijing, China, Aug. 2012, pp. 895–903.

[16] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 83–126, 2019.