IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 10, 2019, accepted May 28, 2019, date of publication June 3, 2019, date of current version June 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2920486

Covariate Shift Adaptation for Structured Regression With Frank–Wolfe Algorithms

SANTOSH V. CHAPANERI, (Member, IEEE), AND DEEPAK J. JAYASWAL
Department of Electronics and Telecommunication Engineering, St. Francis Institute of Technology, University of Mumbai, Mumbai 400103, India
Corresponding author: Santosh V. Chapaneri (santoshchapaneri@sfit.ac.in)

ABSTRACT This paper concerns structured regression problems wherein the issue of covariate shift is addressed, which aims at reducing the discrepancy in training and test data distributions, using computationally efficient and sparse optimization principles. In particular, the projection-free Frank–Wolfe optimization algorithms are used to learn the importance weights and re-weight the training data in the context of covariate shift. To determine the unbiased estimates of the weights, Kullback–Leibler importance estimation procedure is used but its computational cost can be high since it is based on projected gradient optimization. Instead of using the standard Frank–Wolfe algorithm, we adapt its variants and propose away-steps Frank–Wolfe and pairwise Frank–Wolfe covariate shift algorithms to correct the covariate shift. The results highlight the improved computational efficiency and sparsity achieved while learning the importance weights on synthetic as well as benchmark datasets. Furthermore, importance weighted Sharma-Mittal twin Gaussian process structured regression framework is proposed to incorporate the learned weights from covariate shift algorithms, and its equations are derived for importance weighted derivatives and uncertainties. The performance of proposed algorithms is evaluated on two applications of structured regression, namely, human pose estimation and music mood estimation, where the benefit of handling covariate shift is demonstrated with improved performance relative to the state-of-the-art techniques.

INDEX TERMS Covariate shift, Frank-Wolfe optimization, structured regression.

I. INTRODUCTION

Most machine learning methods make the assumption that the training and test data are sampled independently and identically (i.i.d.) from the *same* distribution. However, this assumption is violated in many real-world applications due to which the training data available for learning the model is not an adequate representation of the test data on which the learned model will be ultimately deployed. This concept of covariate shift is defined as follows: $p_{\text{tr}}(\mathbf{x}, \mathbf{y})$ and $p_{\text{te}}(\mathbf{x}, \mathbf{y})$ differ only via $p_{\text{tr}}(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p_{\text{tr}}(\mathbf{x})$ and $p_{\text{te}}(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p_{\text{te}}(\mathbf{x})$, i.e. the conditional probabilities $p(\mathbf{y}|\mathbf{x})$ remain unchanged [1]. A common technique for correcting this covariate shift is to determine the importance weight $w(\mathbf{x}) = \frac{p_{\text{te}}(\mathbf{x})}{p_{\text{tr}}(\mathbf{x})}$ from the training and test data densities. However, estimating these densities is known to be a hard problem as it suffers from the curse of dimensionality and is unreliable for high-dimensional input data [2]. It is thus desirable to learn the importance weight directly without explicitly estimating these densities.

The problem of covariate shift is solved in an unsupervised manner in this work using Frank-Wolfe (FW) algorithms for structured regression framework. FW method is known to efficiently solve constrained convex optimization problems by considering linearization of the objective function (compared to the quadratic solution in conventional gradient methods) with obtaining sparse solutions [3], [4]. Frank-Wolfe covariate shift (FWCS) method was proposed in [5] to learn the importance weights of the Kullback-Leibler Importance Estimation Procedure (KLIEP) [2] by using the standard FW algorithm. To achieve higher sparsity and computational efficiency, we adapt the Away-steps and Pairwise variants of the standard FW algorithm to learn the importance weights in this work. Due to the use of Frank-Wolfe optimization principles, the proposed covariate shift algorithms have a low computational cost per iteration due to solving a linear optimization sub-problem in each iteration. The convergence rate of proposed algorithms is analyzed and an empirical

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

study on benchmark applications is presented to demonstrate the performance with statistical validity.

Structured prediction is an actively studied topic in the machine intelligence community due to its prevalence in real-world applications where the target variables are inter-dependent. For example, in the field of computer vision, 3D human pose estimation [6], [7] is a challenging and important task with several commercial applications (e.g. Microsoft Kinect). Another example in the field of music information retrieval is music mood estimation [8], [9] where the goal is to estimate the 2D mood of music across valence and arousal dimensions to automatically annotate audio clips in a given music library and recommend relevant audio clips for specific mood queries (e.g. Musicovery). In such cases, the output is typically multi-dimensional and thus the goal of structured regression is to learn the mapping $f : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_y}$ directly from multivariate input features $\mathbf{x} \in \mathbb{R}^{d_x}$ to multivariate target variables $\mathbf{y} \in \mathbb{R}^{d_y}$ and exploit the correlation between multivariate data dimensions, rather than inaccurately estimating each dimension of the target variable separately. The structured regression framework is often formulated as an optimization problem that is typically solved using several well-known optimizers such as the BFGS quasi-Newton optimizer.

CONTRIBUTIONS

The contributions of this work are three-fold:

a) The Away-steps and Pairwise variants of the standard Frank-Wolfe algorithm are adapted for correcting the covariate shift based on KLIEP to learn the importance weights and two algorithms, namely Away-steps Frank-Wolfe Covariate Shift (AFWCS) and Pairwise Frank-Wolfe Covariate Shift (PFWCS), are proposed. The update equations for FWCS, AFWCS, and PFWCS algorithms are also derived. The PFWCS algorithm results in sparser iterates and computationally efficient optimum solutions as demonstrated on synthetic as well as benchmark datasets.

b) After estimating the importance weights, we modify the Sharma-Mittal Twin Gaussian Process (SMTGP) structured regression formulation [10] to handle the covariate shift. Its optimization equation, derivatives, and the uncertainty parameter are updated to incorporate the weights of the training data resulting in importance weighted SMTGP (IW-SMTGP).

c) The performance of the proposed covariate shift algorithms and IW-SMTGP is demonstrated on two benchmark applications of structured regression by considering various covariate shift scenarios: (i) Human Pose Estimation and (ii) Music Mood Estimation.

The rest of this paper is organized as follows: Section II reviews related work on covariate shift, advances in Frank-Wolfe optimization methods and structured prediction. Section III briefly describes the KLIEP method and its correction with the standard FW algorithm followed by the proposed FW variant algorithms and their illustrative results. In Section IV, the SMTGP framework is modified to handle the learned weights for covariate shift. Experimental results are presented in Section V using benchmark datasets under various covariate shift scenarios. Finally, the conclusion is given in Section VI with a summary of the contributions.

II. RELATED WORK

A comprehensive overview of dataset shift in classification problems is presented in [1] illustrating the concept of covariate shift along with its causes and applications. The issue of covariate shift is clearly evident when the learned model on training dataset generalizes poorly to novel data. Several techniques are proposed in the literature to solve this problem of covariate shift, e.g. importance estimation based on Kullback-Leibler divergence (KLIEP) [2], least squares importance fitting (LSIF) [11], relative unconstrained LSIF (RuLSIF) [12], KLIEP with Gaussian mixture models (GM-KLIEP) [13], etc. While LSIF and RuLSIF use a closed-form solution that can be analytically obtained, its resulting estimates can be biased [12]. These techniques are primarily unsupervised as they require only the features to estimate the importance weight without depending on the target. In this work, KLIEP is used as the base technique for correcting the covariate shift since its optimization problem is convex resulting in a unique global solution, it has an unbiased estimate, and it can be efficiently solved with FW algorithms.

The FW algorithm typically solves problems of the form $min_{g \in \mathcal{G}} F(g)$ where the function $F : \mathcal{G} \rightarrow \mathbb{R}$ is convex and continuously differentiable and $\mathcal{G}$ is a closed and bounded convex set equipped with inner product $\langle \cdot, \cdot \rangle$. The standard FW algorithm is shown in Algorithm 1 where starting with an initial point $g_0 \in \mathcal{G}$, the algorithm finds the feasible point $s_t \in \mathcal{G}$ that minimizes the linearization of F at the current point g_t. The next iterate g_{t+1} is then updated as a convex combination of g_t and s_t with a suitable step-size ρ_t obtained via line-search. This algorithm converges at an $\mathcal{O}(1/T)$ rate where T is the number of iterations required to achieve convergence of the objective function, and can even achieve an $\mathcal{O}(1/T^2)$ rate when both the objective function and the constraint set are strongly convex [14].

Algorithm 1 Standard Frank-Wolfe

Input: $g_0 \in \mathcal{G}$ **, Output:** Optimum g

procedure StandardFW
  for $t = 0, 1, \ldots$ **do**
    if g_t is a stationary point, **then return** g_t
    Compute $s_t \leftarrow \underset{s \in \mathcal{G}}{\text{argmin}} \langle s, \nabla F(g_t) \rangle$
    Update: $g_{t+1} = (1 - \rho_t) g_t + \rho_t s_t$, for $\rho_t \in [0, 1]$
  end for
end procedure

Jaggi *et al.* in [15] demonstrated several properties of the FW algorithm and its variants to prove linear convergence rates even under relaxed assumptions regarding convexity

of the objective function. The away-steps variant of FW algorithm was proposed in [16] motivated by removing the influence of "bad" visited vertices, and was shown in [17] to converge linearly by relaxing the strongly convex assumption on the objective function. FW algorithms were applied for matrix factorizations in [3] and convergence was shown using duality gap certificates (i.e. approximation quality). A stochastic version of the FW algorithm was proposed in [18] for non-convex optimization problems with improved convergence rates using variance reduction techniques. The optimization problem for image and video co-localization was formulated as a FW problem in [4] leading to increased computational efficiency. In the context of structured prediction, FW method was applied to structured support vector machine (SSVM) in [19] with a block-coordinate setting where the FW duality gap was shown to be equivalent to the Lagrange duality gap of SVM algorithm. In the context of covariate shift, the standard FW algorithm was used in [5] to improve the performance of KLIEP and Kernel Mean Matching (KMM) techniques for estimating the importance weights. To achieve a computationally efficient and sparser solution of KLIEP, we extend this line of work with the known variants of FW algorithm.

In the case of discrete outputs (classification), significant work is reported in the literature for structured prediction, e.g. structural SVM [19], conditional random fields (CRF) [20], Markov random field models [21], etc. Though it is possible to convert regression problems to classification ones, it can lead to loss of information since there is no clarity on the number of classes to be used. Twin Gaussian process (TGP) was first introduced in [22] to solve the structured regression problem by minimizing the Kullback-Leibler (KL) divergence (KLTGP) between the input and output marginal Gaussian Processes (GP) and exploiting the dependencies between multi-dimensional structured output. TGP was applied to human pose estimation and was shown to perform remarkably well relative to conventional GP regression and K-nearest neighbors regression techniques. Since the KL divergence is not a symmetric measure, inverse KLTGP (IKLTGP) was also proposed in [22] to measure the inverse KL divergence between the output and input marginal GPs. A generic version of TGP was proposed in [10] that measures the Sharma-Mittal (SM) divergence between the marginal GPs. KL divergence is a limiting case of the general SM divergence and a closed-form solution of SM divergence for multivariate Gaussian distributions was derived in [23]. Sharma-Mittal TGP (SMTGP) was shown to outperform KLTGP for the application of human pose estimation while having the same quadratic computational complexity as that of KLTGP. In this work, we thus modify the SMTGP framework to incorporate the importance weights and handle various covariate shift scenarios.

III. FRANK-WOLFE COVARIATE SHIFT ALGORITHMS

In this section, the KLIEP importance estimation procedure is discussed briefly along with its adaptation by the Frank-Wolfe concept, i.e. FWCS (Frank-Wolfe Covariate Shift) proposed in [5]. We propose two new algorithms for estimating weights due to covariate shift based on the variants of the Frank-Wolfe concept, namely Away-steps (AFWCS) and Pairwise (PFWCS) algorithms, and analyze their convergence rates. The performance of these algorithms is demonstrated on a synthetic dataset.

A. IMPORTANCE ESTIMATION PROCEDURE AND ITS FRANK-WOLFE ADAPTATION

A popular covariate shift method, namely KLIEP (Kullback-Leibler Importance Estimation Procedure) proposed in [2] determines the importance estimate $w(\mathbf{x}) = \hat{p}_{\text{te}}(\mathbf{x})/p_{\text{tr}}(\mathbf{x})$ such that the Kullback-Leibler (KL) divergence from the true density $p_{\text{te}}(\mathbf{x})$ to its estimate $\hat{p}_{\text{te}}(\mathbf{x})$ is minimized without explicitly computing the densities. The weights are parameterized as mixtures of Gaussians and modeled as (1), where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{n_{\text{te}}}]^\intercal$ are the mixing coefficients, $\kappa(\cdot)$ is the kernel function, n_{tr} and n_{te} are the number of training and testing data samples, respectively, and $[\cdot]^\intercal$ denotes vector transpose. The training samples are re-weighted by $w(\mathbf{x}^{\text{tr}})$ to reduce the discrepancy between the training and test data distributions.

$$w(\mathbf{x}^{\text{tr}}) = \sum_{l=1}^{n_{\text{te}}} \alpha_l \kappa_l(\mathbf{x}^{\text{tr}}) = \sum_{l=1}^{n_{\text{te}}} \alpha_l \exp\left(\frac{-\|\mathbf{x}^{\text{tr}} - \mathbf{x}_l^{\text{te}}\|^2}{2\sigma^2}\right) \tag{1}$$

The optimization problem of KLIEP is given by (2), which is convex resulting in a unique global solution obtained by the *projected gradient method* [2]. When the number of test samples n_{te} is high (e.g. in the cases of large-scale data), the authors propose to use only a subset of testing data as Gaussian centers to reduce the computational cost of finding the projections. Additionally, likelihood cross-validation can be used for model selection of the optimum σ value, however, this results in biased estimates under covariate shift. Thus, to determine the σ parameter used in the kernel function of KLIEP, the unbiased importance weighted variant of cross-validation (IWCV) [24] is used.

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}) = \sum_{j=1}^{n_{\text{te}}} \log\left(\sum_{l=1}^{n_{\text{te}}} \alpha_l \kappa_l(\mathbf{x}_j^{\text{te}})\right) \\ &\text{s.t.} \sum_{i=1}^{n_{\text{tr}}} \sum_{j=1}^{n_{\text{te}}} \alpha_j \kappa_j(\mathbf{x}_i^{\text{tr}}) = n_{\text{tr}}; \quad \alpha_1, \alpha_2, \ldots \alpha_{n_{\text{te}}} \geq 0 \end{aligned} \tag{2}$$

In [5], authors used the standard Frank-Wolfe algorithm to solve the optimization problem of KLIEP resulting in a sparser solution relative to the original KLIEP method. The Frank-Wolfe algorithm (also known as the *conditional gradient* or *projection-free method*) iteratively approximates the objective function linearly by using linear programming to choose the ascent direction. To use the FW algorithm for KLIEP, the gradient $\mathbf{g} = [g_1, \ldots, g_{n_{\text{te}}}]^\intercal$ of the objective function given by (3) is used to solve the linear maximization problem. From the constraint on $\boldsymbol{\alpha}$, its upper bound

$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{n_{\text{te}}}]^\intercal$ is given by (4), i.e. $0 \leq \alpha_l \leq \beta_l$, for $l = 1, \ldots, n_{\text{te}}$.

$$g_l = \frac{\partial F(\boldsymbol{\alpha})}{\partial \alpha_l} = \sum_{j=1}^{n_{\text{te}}} \frac{\kappa_l(\mathbf{x}_j^{\text{te}})}{\sum_{l'=1}^{n_{\text{te}}} \alpha_{l'} \kappa_{l'}(\mathbf{x}_j^{\text{te}})} \tag{3}$$

$$\beta_l = \frac{n_{\text{tr}}}{\sum_{i=1}^{n_{\text{tr}}} \kappa_l(\mathbf{x}_i^{\text{tr}})} \tag{4}$$

The pseudo-code of Frank-Wolfe Covariate Shift (FWCS) for KLIEP is given in Algorithm 2. The training and test data samples are given as input and the output is the importance weight for each training sample. In line 2, the upper bound of $\boldsymbol{\alpha}$ is calculated and the iteration counter t is set to 0. The current iterate $\boldsymbol{\alpha}_t$ can be expressed as an atomic decomposition $\boldsymbol{\alpha}_t = \sum_{l=1}^{n_{\text{te}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)}$ such that $\sum_{l=1}^{n_{\text{te}}} \boldsymbol{\mu}_t(l) = 1$ and $\boldsymbol{\mu}_t(l) \geq 0$, where $\boldsymbol{\beta}^{(l)} = \boldsymbol{\beta} \odot \mathbf{e}^{(l)}$ with $\mathbf{e}^{(l)}$ as the unit basis vector with all entries 0 except 1 at l, and $\odot$ denotes the element-wise (Hadamard) product. The active set $\mathcal{S}_t = \{l : \boldsymbol{\mu}_t(l) \neq 0\}$ contains the non-zero atom locations visited up to iteration t. In line 3, the mixing coefficients $\boldsymbol{\alpha}$, atoms $\boldsymbol{\mu}$ and the active set $\mathcal{S}$ are initialized, where i_β is the index of the minimum element of the upper bound $\boldsymbol{\beta}$. Note that initializing $\boldsymbol{\mu}$ and $\mathcal{S}$ is not explicitly required in the FWCS algorithm; however, we use it here to relate this algorithm with the proposed algorithms in the next sub-section. Lines $4 - 12$ are iterated till convergence of the objective function F. At each iteration t, the gradient $\mathbf{g}_t$ of F is calculated and the location l_t^{FW} is obtained as the largest coordinate of the element-wise product of the gradient and the upper bound. The towards (ascent) direction $\mathbf{d}_t^{\text{FW}}$ is obtained via the FW linear maximization principle in line 6. The duality gap computed by the inner product $\langle \mathbf{g}_t, \mathbf{d}_t^{\text{FW}} \rangle$ associated with the objective function F at the current iteration t can be utilized as a measure of proximity to the optimum solution.

Using line search, the step-size ρ_t is determined to move in the towards direction. Instead of using an exact line search method (requiring iterative optimization that may converge only asymptotically), an inexact line search is performed using the classical Armijo rule. The pseudo-code of Armijo (backtracking) line search to maximize the objective function is given in Algorithm 3. The line search is iterated until the Armijo condition stated in Theorem 1 (in Appendix A) is satisfied to ensure a sufficient increase in the value of the objective function F. This line search method is known to have a global linear rate of convergence [25, Ch. 2]. The mixing coefficients $\boldsymbol{\alpha}_{t+1}$ are then updated along the towards direction in line 8. The atoms $\boldsymbol{\mu}_{t+1}$ and the active set $\mathcal{S}_{t+1}$ are updated in lines 9 and 10 respectively, as derived in Appendix B.1. Finally, in line 13, the importance weights $\mathbf{w}$ are estimated using the converged sequence of $\boldsymbol{\alpha}_t$ with (1).

Algorithm 2 Frank-Wolfe Covariate Shift (FWCS) for KLIEP

Input: $\{\mathbf{x}_i^{\text{tr}}\}_{i=1}^{n_{\text{tr}}}, \{\mathbf{x}_j^{\text{te}}\}_{j=1}^{n_{\text{te}}}$ **, Output:** $\mathbf{w} = \{w(\mathbf{x}_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$

1: **procedure** FWCS
2: Compute upper bound $\boldsymbol{\beta}$ (4) and set $t \leftarrow 0$
3: Initialize: $\boldsymbol{\alpha}_0 \leftarrow \mathbf{0}, i_\beta \leftarrow \underset{i}{\text{argmin}}(\boldsymbol{\beta})_i, \boldsymbol{\alpha}_0(i_\beta) \leftarrow \boldsymbol{\beta}(i_\beta), \boldsymbol{\mu}_0 \leftarrow \mathbf{0}, \boldsymbol{\mu}_0(i_\beta) \leftarrow 1, \mathcal{S}_0 \leftarrow \{i_\beta\}$
4: **repeat**
5: Compute gradient $\mathbf{g}_t$ (3) ▷ using previous $\boldsymbol{\alpha}_t$
6: Find $l_t^{\text{FW}} \leftarrow \underset{l}{\text{argmax}}\ (\mathbf{g}_t \odot \boldsymbol{\beta})_l$;
Compute $\mathbf{d}_t^{\text{FW}} \leftarrow \boldsymbol{\beta}^{(l_t^{\text{FW}})} - \boldsymbol{\alpha}_t$ ▷ Towards direction
7: Find $\rho_t \leftarrow \underset{\rho \in [0,1]}{\text{argmax}}\ F(\boldsymbol{\alpha}_t + \rho \mathbf{d}_t^{\text{FW}})$ ▷ Line search
8: Update: $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\text{FW}}$
9: Update: $\boldsymbol{\mu}_{t+1} \leftarrow (1 - \rho_t)\boldsymbol{\mu}_t$;
$\boldsymbol{\mu}_{t+1}(l_t^{\text{FW}}) \leftarrow \boldsymbol{\mu}_{t+1}(l_t^{\text{FW}}) + \rho_t$
10: Update: $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{l_t^{\text{FW}}\}$ ▷ Active set
11: $t \leftarrow t + 1$
12: **until** convergence of (2)
13: Compute weights $\mathbf{w}$ with $\boldsymbol{\alpha}_t$ (1)
14: **end procedure**

Algorithm 3 Armijo Line Search

Input: $\boldsymbol{\alpha}_t, \mathbf{d}_t, F, \mathbf{g}_t, \rho_{max}, \tau \in (0, 1), \xi \in (0, 1)$**, Output:** ρ_t

1: **procedure** ArmijoLineSearch
2: Set $\rho \leftarrow \rho_{max}$
3: **while** $F(\boldsymbol{\alpha}_t + \rho \mathbf{d}_t) < F(\boldsymbol{\alpha}_t) + \tau\rho\langle \mathbf{g}_t, \mathbf{d}_t \rangle$ **do**
4: $\rho \leftarrow \xi\rho$
5: **end while**
6: $\rho_t \leftarrow \rho$
7: **end procedure**

B. PROPOSED FRANK-WOLFE COVARIATE SHIFT ALGORITHMS

The sequence of iterates produced by the FW algorithm converges to the optimal value linearly when the optimal solution lies in the interior of the feasible set of a polytope. Otherwise, the convergence rate is sub-linear due to zig-zagging effects [15]. To guarantee linear convergence rate, the variants of Frank-Wolfe algorithms, namely Away-steps FW and Pairwise FW algorithms are proposed in the literature [15], [16], [26], [27]. Jaggi *et al.* in [15] studied the variants of FW algorithms in detail and derived global linear convergence rate for all its variants.

Figure 1 illustrates the directions taken by the FW algorithm and its variants from the current solution $\boldsymbol{\alpha}_t$, where $\boldsymbol{\alpha}^*$ is the optimum solution. In the standard FW algorithm, the towards direction $\mathbf{d}_t^{\text{FW}}$ is always chosen during each iteration by finding the location l_t^{FW} that maximizes the potential of ascent; however, this can cause zig-zagging when the optimal solution lies closer to the boundary of the polytope and thus may need more iterations to converge [15]. To address this, away-steps can be taken by moving away along $\mathbf{d}_t^{\text{AFW}}$. The away direction is obtained by finding the location l_t^{AFW} that minimizes the potential of descent. In the pairwise FW algorithm, the idea is to move pairwise from the

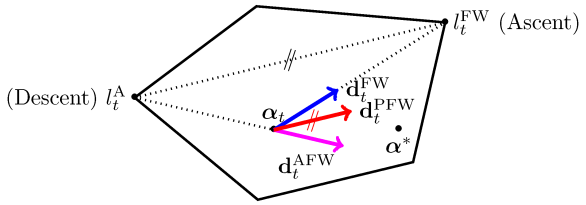

FIGURE 1. Illustration of Frank-Wolfe towards, away and pairwise directions.

away direction to the towards direction in the same iteration resulting in $\mathbf{d}_t^{\mathrm{PFW}} = \mathbf{d}_t^{\mathrm{FW}} + \mathbf{d}_t^{\mathrm{AFW}}$. The pairwise FW (PFW) algorithm outperforms both standard FW as well as away-step FW (AFW) algorithms, especially in the case where sparser solutions can be found [15]. We modify the FWCS algorithm to consider the *away* as well as *pairwise* directions for handling the covariate shift and propose AFWCS and PFWCS algorithms to estimate the importance weights $\mathbf{w}$.

1) AWAY-STEPS FRANK-WOLFE COVARIATE SHIFT

The pseudo-code of Away-steps FWCS (AFWCS) is given in Algorithm 4. The working is similar to Algorithm 2 with the following modifications. Besides finding the towards direction $\mathbf{d}_t^{\mathrm{FW}}$ in line 6, the away direction $\mathbf{d}_t^{\mathrm{AFW}}$ is also found in line 7; however, the search for the location l_t^{AFW} (obtained as the smallest coordinate of the element-wise product of the gradient and the upper bound) is over only the typically smaller active set $\mathcal{S}_t$, which makes it fundamentally easier than finding l_t^{FW}. The maximum step-size $\rho_{\max}$ (derived in Appendix B.2) given in line 8 guarantees the feasibility of the solution in the away direction. In line 9, if the duality gap in the towards direction is higher compared to that of the away direction, then the towards direction is chosen; otherwise, the away direction is chosen. In the case of towards direction, the working is identical to that of the FWCS algorithm. However, in the case of away direction, the line search is conducted over the range $[0, \rho_{\max}]$ to ensure feasibility in line 15 and the mixing coefficients $\boldsymbol{\alpha}_{t+1}$ are updated along $\mathbf{d}_t^{\mathrm{AFW}}$ in line 16 followed by updating the atoms $\boldsymbol{\mu}_{t+1}$ in line 17. In line 18, if the step-size obtained via the line search is same as the maximum step-size $\rho_{\max}$, this is referred to as a *drop step*, as it fully removes the location l_t^{AFW} from the current active set $\mathcal{S}_t$ by setting its atom to zero. The update equations for atoms $\boldsymbol{\mu}_{t+1}$ and active set $\mathcal{S}_{t+1}$ are derived in Appendix B.2.

2) PAIRWISE FRANK-WOLFE COVARIATE SHIFT

The pseudo-code of Pairwise FWCS (PFWCS) is given in Algorithm 5. The working is similar to Algorithm 4 with the following modifications. Instead of choosing the towards or away direction, the pairwise direction is always used at each iteration t in line 9. That is, the solution moves away from the away location l_t^{AFW} and also gets closer to the towards location l_t^{FW} in the *same* iteration. The pairwise

Algorithm 4 Away-Steps Frank-Wolfe Covariate Shift (AFWCS) for KLIEP

Input: $\{\mathbf{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}, \{\mathbf{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, **Output:** $\mathbf{w} = \{w(\mathbf{x}_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$

1: **procedure** AFWCS
2: Compute upper bound $\boldsymbol{\beta}$ (4) and set $t \leftarrow 0$
3: Initialize: $\boldsymbol{\alpha}_0 \leftarrow \mathbf{0}, i_\beta \leftarrow \underset{i}{\mathrm{argmin}}(\boldsymbol{\beta})_i, \boldsymbol{\alpha}_0(i_\beta) \leftarrow \boldsymbol{\beta}(i_\beta), \boldsymbol{\mu}_0 \leftarrow \mathbf{0}, \boldsymbol{\mu}_0(i_\beta) \leftarrow 1, \mathcal{S}_0 \leftarrow \{i_\beta\}$
4: **repeat**
5: Compute gradient $\mathbf{g}_t$ (3) ▷ using previous $\boldsymbol{\alpha}_t$
6: Find $l_t^{\mathrm{FW}} \leftarrow \underset{l}{\mathrm{argmax}}\,(\mathbf{g}_t \odot \boldsymbol{\beta})_l$;
Compute $\mathbf{d}_t^{\mathrm{FW}} \leftarrow \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\alpha}_t$ ▷ Towards direction
7: Find $l_t^{\mathrm{AFW}} \leftarrow \underset{l \in \mathcal{S}_t}{\mathrm{argmin}}\,(\mathbf{g}_t \odot \boldsymbol{\beta})_l$;
Compute $\mathbf{d}_t^{\mathrm{AFW}} \leftarrow \boldsymbol{\alpha}_t - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}$ ▷ Away direction
8: Compute $\rho_{\max} \leftarrow \frac{\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})}{1-\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})}$ ▷ Max. step of away direction
9: **if** $\langle \mathbf{g}_t, \mathbf{d}_t^{\mathrm{FW}} \rangle \geq \langle \mathbf{g}_t, \mathbf{d}_t^{\mathrm{AFW}} \rangle$ **then** ▷ Choose direction
10: Find $\rho_t \leftarrow \underset{\rho \in [0,1]}{\mathrm{argmax}}\, F(\boldsymbol{\alpha}_t + \rho \mathbf{d}_t^{\mathrm{FW}})$ ▷ Towards step
11: Update: $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\mathrm{FW}}$
12: Update: $\boldsymbol{\mu}_{t+1} \leftarrow (1-\rho_t)\boldsymbol{\mu}_t$;
$\boldsymbol{\mu}_{t+1}(l_t^{\mathrm{FW}}) \leftarrow \boldsymbol{\mu}_{t+1}(l_t^{\mathrm{FW}}) + \rho_t$
13: Update: $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{l_t^{\mathrm{FW}}\}$
14: **else**
15: Find $\rho_t \leftarrow \underset{\rho \in [0, \rho_{\max}]}{\mathrm{argmax}}\, F(\boldsymbol{\alpha}_t + \rho \mathbf{d}_t^{\mathrm{AFW}})$ ▷ Away step
16: Update: $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\mathrm{AFW}}$
17: Update: $\boldsymbol{\mu}_{t+1} \leftarrow (1+\rho_t)\boldsymbol{\mu}_t$;
$\boldsymbol{\mu}_{t+1}(l_t^{\mathrm{AFW}}) \leftarrow \boldsymbol{\mu}_{t+1}(l_t^{\mathrm{AFW}}) - \rho_t$
18: **if** $(\rho_t == \rho_{\max})$ **do**
$\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \setminus \{l_t^{\mathrm{AFW}}\}$, $\boldsymbol{\mu}_{t+1}(l_t^{\mathrm{AFW}}) \leftarrow 0$ ▷ Drop step
19: **end if**
20: $t \leftarrow t+1$
21: **until** convergence of (2)
22: Compute weights $\mathbf{w}$ with $\boldsymbol{\alpha}_t$ (1)
23: **end procedure**

direction can be obtained as a superposition of the towards and away steps:

$$\mathbf{d}_t^{\mathrm{PFW}} = \underbrace{\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\alpha}_t}_{\text{(towards step)}} + \underbrace{\boldsymbol{\alpha}_t - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}}_{\text{(away step)}}$$
$$\therefore \mathbf{d}_t^{\mathrm{PFW}} = \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} = \mathbf{d}_t^{\mathrm{FW}} + \mathbf{d}_t^{\mathrm{AFW}} \quad (5)$$

Thus, the pairwise direction $\mathbf{d}_t^{\mathrm{PFW}}$ is given by (5). In line 8, the maximum step-size $\rho_{\max}$ (derived in Appendix B.3) ensures the feasibility of the current solution using line search in line 10. Due to the pairwise direction, only the atoms

Algorithm 5 Pairwise Frank-Wolfe Covariate Shift (PFWCS) for KLIEP

…as in Algorithm 4, except replacing lines 8 to 19 by:

8: Compute $\rho_{\max} \leftarrow \boldsymbol{\mu}_t(l_t^{\text{AFW}})$ ▷ Max. step of away direction

9: Compute $\mathbf{d}_t^{\text{PFW}} \leftarrow \mathbf{d}_t^{\text{FW}} + \mathbf{d}_t^{\text{AFW}}$ ▷ Pairwise direction

10: Find $\rho_t \leftarrow \underset{\rho \in [0, \rho_{\max}]}{\text{argmax}}\ F(\boldsymbol{\alpha}_t + \rho \mathbf{d}_t^{\text{PFW}})$ ▷ Line search

11: Update: $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\text{PFW}}$

12: Update: $\boldsymbol{\mu}_{t+1} \leftarrow \boldsymbol{\mu}_t$;
$\boldsymbol{\mu}_{t+1}(l_t^{\text{FW}}) \leftarrow \boldsymbol{\mu}_t(l_t^{\text{FW}}) + \rho_t$;
$\boldsymbol{\mu}_{t+1}(l_t^{\text{AFW}}) \leftarrow \boldsymbol{\mu}_t(l_t^{\text{AFW}}) - \rho_t$

13: Update: $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{l : \boldsymbol{\mu}_{t+1}(l) \neq 0\}$

14: **if** $(\rho_t == \rho_{\max})$ **do** ▷ Modify the active set
if $(l_t^{\text{FW}} \in \mathcal{S}_t)$ **do**
$\mathcal{S}_{t+1} \leftarrow \mathcal{S}_{t+1} \setminus \{l_t^{\text{AFW}}\},\ \boldsymbol{\mu}_{t+1}(l_t^{\text{AFW}}) \leftarrow 0$ ▷ Drop step
else
$\mathcal{S}_{t+1} \leftarrow \mathcal{S}_{t+1} \cup \{l_t^{\text{FW}}\} \setminus \{l_t^{\text{AFW}}\}$ ▷ Swap step

related to l_t^{FW} and l_t^{AFW} are updated in line 12, leaving other atoms unchanged (derived in Appendix B.3). Note that, in contrast, the FWCS and AFWCS algorithms updates *all* active atoms in each iteration. The active set is updated to contain all locations with non-zero atom weights in line 13. In line 14, we can have either a *drop step* or a *swap step* depending on certain conditions outlined as follows. An iteration t is a

(i) *drop step*, if $\rho_t = \rho_{\max} < 1$ and $|\mathcal{S}_{t+1}| = |\mathcal{S}_t| - 1$,
(ii) *swap step*, if $\rho_t = \rho_{\max} < 1$ and $|\mathcal{S}_{t+1}| = |\mathcal{S}_t|$, and
(iii) *good step*, if it is neither a *drop* nor a *swap step*.

The covariate shift algorithms can take *good/drop/swap* steps as follows:

(i) FWCS algorithm can have only *good steps*, since $\rho_{\max} = 1$ at each iteration (refer Appendix B.1).
(ii) AFWCS algorithm can have *good steps* as well as *drop steps* (when the away direction is chosen), but *swap steps* can never occur:
  – *good step*, if $\rho_{\max} = 1$ (i.e. $\mathbf{d}_t = \mathbf{d}_t^{\text{FW}}$), or $\rho_{\max} < 1$ (i.e. $\mathbf{d}_t = \mathbf{d}_t^{\text{AFW}}$) and $\rho_t < \rho_{\max}$,
  – *drop step*, if $\rho_{\max} < 1$ (i.e. $\mathbf{d}_t = \mathbf{d}_t^{\text{AFW}}$) and $\rho_t = \rho_{\max}$ (the atom at location l_t^{AFW} is set to 0 and this location is removed from the active set).
(iii) PFWCS algorithm can have *good steps* as well as *drop* or *swap steps*:
  – *good step*, if $\rho_{\max} = 1$, or $\rho_{\max} < 1$ and $\rho_t < \rho_{\max}$,
  – *drop step*, if $\rho_t = \rho_{\max} < 1$ and $l_t^{\text{FW}} \in \mathcal{S}_t$ (the atom at location l_t^{AFW} is set to 0 and this location is removed from the active set),
  – *swap step*, if $\rho_t = \rho_{\max} < 1$ and $l_t^{\text{FW}} \notin \mathcal{S}_t$ (the location l_t^{AFW} is swapped with l_t^{FW} in the active set).

C. CONVERGENCE RATE ANALYSIS

Let $h_t = F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*)$ be the sub-optimality error for an optimal solution $\boldsymbol{\alpha}^*$ and let $k(t)$ be the number of *good steps* taken till iteration t. The sub-optimality error h_t decreases geometrically at a linear convergence rate $h_{t+1} \leq (1 - \nu)h_t$, where ν is a constant depending on the curvature constant C_F (related to the second-order derivative of F) and the geometric strong convexity constant μ_F [26]. The number of *good steps* are bounded by $k(t) = t$ for FWCS, $k(t) \geq t/2$ for AFWCS and $k(t) \geq t/(3|\mathcal{A}|! + 1)$ for PFWCS, where $\mathcal{A} \subseteq \mathbb{R}^d$ is the finite set of atoms. This results in the global linear convergence rate of $h_t \leq h_0 \exp(-\nu k(t))$ for all three covariate shift algorithms, leading to the computational cost of $\mathcal{O}(1/t)$. Also, the number of *drop steps* in both AFWCS and PFWCS algorithms is bounded by $t/2$ and the maximum number of *swap steps* between any two *good steps* in PFWCS algorithm is bounded by $3|\mathcal{A}|!$. These results are proved in [15] and are shown to hold true even when the objective function is not globally strongly convex.

A significant advantage of FWCS, AFWCS and PFWCS algorithms is that *all* test data can be used as Gaussian centers to determine the optimal solution since only one Gaussian is activated per iteration (via the FW linear maximization/minimization principle) resulting in efficient and sparser solutions. On the contrary, in the original KLIEP algorithm, a randomly chosen *subset* of test data is used as Gaussian centers (e.g. 100 test data points), which may not be appropriate to model the discrepancy in distributions between the training and test data [5]. Further, the proposed AFWCS and PFWCS algorithms produce sparser mixing coefficients compared to KLIEP and FWCS algorithms as illustrated in Section III-D.

D. ILLUSTRATIVE RESULTS OF COVARIATE SHIFT ALGORITHMS

The performance of proposed algorithms is demonstrated on synthetic data generated from $y = -2x^3 + 3\,\text{sinc}(x) + 1 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.1^2)$, similar to [5]. 500 training points are generated from $\mathcal{N}(0.5, 0.5^2)$ and 300 testing points are generated from $\mathcal{N}(0, 0.3^2)$ to simulate the covariate shift. Figure 2(a), 2(b), 2(c) and 2(d) show the results of KLIEP as well as all three Frank-Wolfe covariate shift algorithms where we observe that they obtain similar weights $\mathbf{w}$; however, the mixing coefficients $\boldsymbol{\alpha}$ of FW algorithms vary significantly as seen in Figure 2(e) (the scale on y-axis is different to highlight the range of values in each case). Out of 300 test points, the original KLIEP algorithm produces 180 non-sparse mixing coefficients $\boldsymbol{\alpha}$, whereas FWCS, AFWCS and PFWCS algorithms produces 73, 18 and 8 non-sparse mixing coefficients, respectively. The pairwise covariate shift (PFWCS) algorithm results in the most sparse solution.

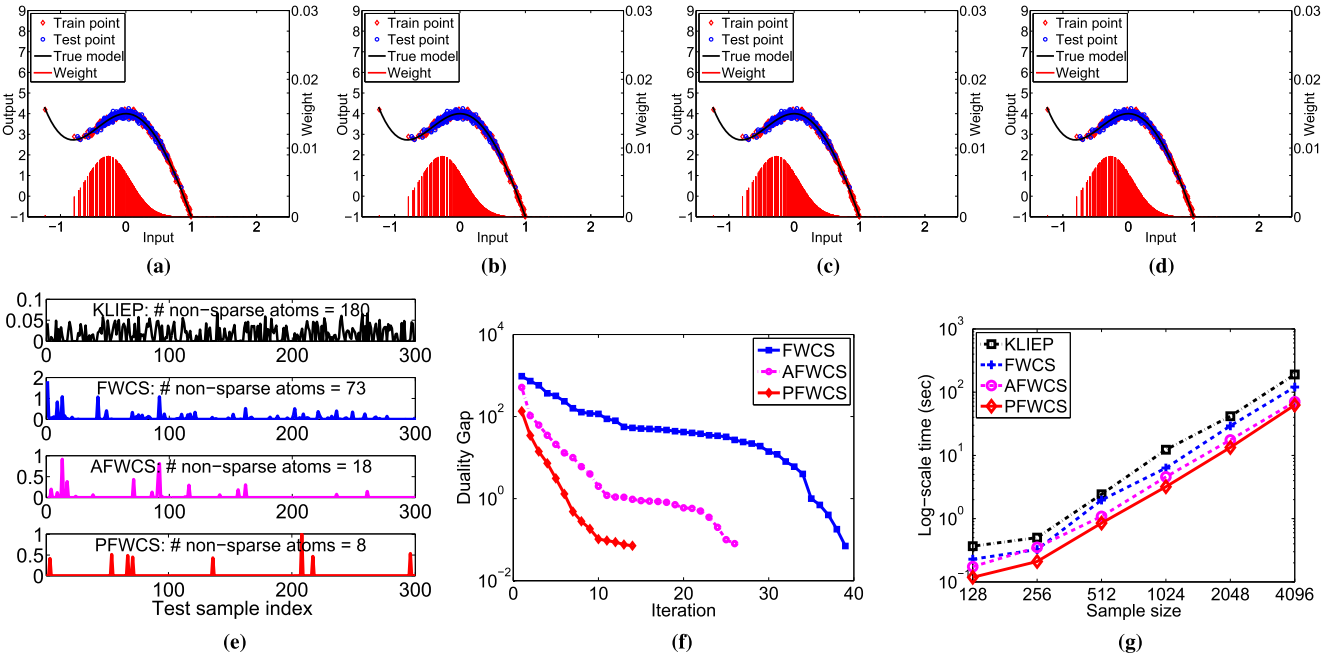

FIGURE 2. Performance of Frank-Wolfe covariate shift algorithms on a synthetic dataset. (a) KLIEP weights, (b) FWCS weights, (c) AFWCS weights, (d) PFWCS weights, (e) Sparsity of α, (F) Duality gap $\langle g_t, d_t \rangle$, (g) Run-time to determine w.

Figure 2(f) shows the decrease in duality gap $\langle \mathbf{g}_t, \mathbf{d}_t \rangle$ with respect to the number of iterations t for all the three algorithms, where we observe that the PFWCS algorithm converges relatively faster and its duality gap is the smallest. Figure 2(g) shows the run-time in log-scale with varying sampling size ($n_{\text{tr}} = n_{\text{te}}$) of the synthetic data. In this case, it is observed that the Frank-Wolfe algorithms are consistently faster than the original KLIEP algorithm and that the PFWCS algorithm converges faster than other algorithms. These algorithms are further evaluated on the benchmark datasets in Section V.

IV. ADAPTING COVARIATE SHIFT FOR STRUCTURED REGRESSION

The conventional Gaussian Process Regression (GPR) technique [28] considers a linear model of the form given by (6), where $\varepsilon^{(d)} \sim \mathcal{N}(0, \sigma^2), f^{(d)}(\mathbf{x}) = \psi^{(d)\intercal}\zeta(\mathbf{x})$ with zero-mean Gaussian prior $\psi^{(d)} \sim \mathcal{N}(\mathbf{0}_p, \Sigma_p)$ and $\zeta(\mathbf{x})$ function maps the d_x-dimensional input vector $\mathbf{x}$ to a p-dimensional feature space. However, it does not capture the dependencies that may exist in the multi-dimensional (structured) output, and hence the Twin Gaussian Process (TGP) framework can be used that explicitly handles the correlation between the output dimensions.

$$y^{(d)} = f^{(d)}(\mathbf{x}) + \varepsilon^{(d)} \tag{6}$$

KLTGP was proposed in [22] to capture the correlations among both multi-dimensional inputs as well as multi-dimensional outputs by minimizing the KL divergence between the input and output GPs. A generic version of TGP known as Sharma-Mittal TGP (SMTGP) using the generalized Sharma-Mittal divergence measure [10] having two parameters (θ, γ) was shown to outperform KLTGP on several benchmark datasets. In this work, we modify SMTGP to incorporate the importance weights learned with the algorithms in Section III for handling the covariate shift. A brief description of SMTGP is given followed by its modification for covariate shift adaptation.

A. SHARMA-MITTAL TWIN GAUSSIAN PROCESS

The joint distributions of input and output data are given by $p(\mathbf{X}, \mathbf{x}) = \mathcal{N}_{\mathbf{X}}(\mathbf{0}, \mathbf{K}_{\mathbf{X}\cup\mathbf{x}})$ and $p(\mathbf{Y}, \mathbf{y}) = \mathcal{N}_{\mathbf{Y}}(\mathbf{0}, \mathbf{K}_{\mathbf{Y}\cup\mathbf{y}})$, where the joint kernel $(N+1) \times (N+1)$ matrices are defined in (7) with $\mathbf{x} \in \mathbb{R}^{d_x}$ as the new test point corresponding to the unknown (multi-dimensional) output $\mathbf{y} \in \mathbb{R}^{d_y}$. The training set comprises of $\mathbf{X} \in \mathbb{R}^{(N\times d_x)}$ and $\mathbf{Y} \in \mathbb{R}^{(N\times d_y)}$ matrices with $N(= n_{\text{tr}})$ data samples $\{\mathbf{x}_i^{\text{tr}}, \mathbf{y}_i^{\text{tr}}\}_{i=1}^{n_{\text{tr}}}$.

$$\mathbf{K}_{\mathbf{X}\cup\mathbf{x}} = \begin{bmatrix} \mathbf{K}_{\mathbf{X}} & \mathbf{k}_{\mathbf{X}}^{\mathbf{x}} \\ \mathbf{k}_{\mathbf{X}}^{\mathbf{x}\intercal} & k_{\mathbf{X}}(\mathbf{x},\mathbf{x}) \end{bmatrix}, \quad \mathbf{K}_{\mathbf{Y}\cup\mathbf{y}} = \begin{bmatrix} \mathbf{K}_{\mathbf{Y}} & \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}} \\ \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}\intercal} & k_{\mathbf{Y}}(\mathbf{y},\mathbf{y}) \end{bmatrix} \tag{7}$$

$\mathbf{K}_{\mathbf{X}}$ is a $N \times N$ kernel matrix consisting of similarity elements $(\mathbf{K}_{\mathbf{X}})_{ij}$ between $\mathbf{x}_i^{\text{tr}}$ and $\mathbf{x}_j^{\text{tr}}$ training input points, $\mathbf{k}_{\mathbf{X}}^{\mathbf{x}}$ is a $N \times 1$ column vector having elements $(\mathbf{k}_{\mathbf{X}}^{\mathbf{x}})_i = \mathbf{K}_{\mathbf{X}}(\mathbf{x}_i^{\text{tr}}, \mathbf{x})$, and $k_{\mathbf{X}}(\mathbf{x},\mathbf{x}) = \mathbf{K}_{\mathbf{X}}(\mathbf{x},\mathbf{x})$ is a scalar. Similar definitions hold for $\mathbf{K}_{\mathbf{Y}}, \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}}$, and $k_{\mathbf{Y}}(\mathbf{y},\mathbf{y})$. Using Gaussian-RBF kernel functions, the similarity kernels for input and output are given by (8), where ρ_x and ρ_y correspond to the kernel bandwidths, λ_x and λ_y are the regularization parameters to avoid over-fitting, and δ is the Kronecker delta function with $\delta_{ij} = 1$ if $i = j$, and 0

otherwise.

$$(\mathbf{K_X})_{ij} = \exp\left(-\frac{\|\mathbf{x}_i^{\text{tr}} - \mathbf{x}_j^{\text{tr}}\|^2}{2\rho_x^2}\right) + \lambda_x \delta_{ij}$$
$$(\mathbf{K_Y})_{ij} = \exp\left(-\frac{\|\mathbf{y}_i^{\text{tr}} - \mathbf{y}_j^{\text{tr}}\|^2}{2\rho_y^2}\right) + \lambda_y \delta_{ij} \quad (8)$$

The SMTGP optimization function $L_{\theta,\gamma}(p(\mathbf{X},\mathbf{x}), p(\mathbf{Y},\mathbf{y}))$ computed with the Sharma-Mittal divergence is given by (9), and its predicted output $\hat{\mathbf{y}}$ is given by (10). Here, $k^{\theta}_{\mathbf{XY}} = (1-\theta)k_{\mathbf{X}}(\mathbf{x},\mathbf{x}) + \theta k_{\mathbf{Y}}(\mathbf{y},\mathbf{y})$ is a scalar, $\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY}} = (1-\theta)\mathbf{k}^{\mathbf{x}}_{\mathbf{X}} + \theta\mathbf{k}^{\mathbf{y}}_{\mathbf{Y}}$ is a $N \times 1$ column vector and $\mathbf{K_{XY}} = (1-\theta)\mathbf{K_X} + \theta\mathbf{K_Y}$ is a $N \times N$ matrix. Similar to KLTGP, the SMTGP problem can be solved using a second-order BFGS quasi-Newton optimizer with line search for optimal step size selection and has a computational complexity of $\mathcal{O}(N^2)$ at test time, where N is the number of data points. This is because $\mathbf{K_X^{-1}}$, $\mathbf{K_Y^{-1}}$ and $\mathbf{K_{XY}^{-1}}$ are pre-computed and stored as they depend only on the training data.

$$L_{\theta,\gamma} = \frac{1}{\gamma - 1}\left[\left(\frac{|\mathbf{K_{X\cup x}}|^{1-\theta}|\mathbf{K_{Y\cup y}}|^{\theta}}{|(1-\theta)\mathbf{K_{X\cup x}} + \theta\mathbf{K_{Y\cup y}}|}\right)^{\frac{(1-\gamma)}{2(1-\theta)}} - 1\right] \quad (9)$$

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\text{argmin}}\ \frac{1}{\gamma - 1}\left[\left(k_{\mathbf{Y}}(\mathbf{y},\mathbf{y}) - \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y}}^{-1}\mathbf{k}_{\mathbf{Y}}^{\mathbf{y}}\right)^{\frac{\theta(1-\gamma)}{2(1-\theta)}} \times \left(k^{\theta}_{\mathbf{XY}} - \mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY}}\mathbf{K}^{-1}_{\mathbf{XY}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY}}\right)^{\frac{-(1-\gamma)}{2(1-\theta)}}\right] \quad (10)$$

The determinants of (7) are given by $|\mathbf{K_{X\cup x}}| = |\mathbf{K_X}| \times \eta_{\mathbf{x}}$ and $|\mathbf{K_{Y\cup y}}| = |\mathbf{K_Y}| \times \eta_{\mathbf{y}}$, where $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{y}}$ are the Schur complements of the joint kernels (refer Appendix C) and can be interpreted as the ratio by which the uncertainty measure (variance) of Gaussian Process changes with respect to the new data point $\mathbf{x}$. Both $\eta_{\mathbf{x}}$ and $\eta_{\mathbf{y}}$ are upper bounded by $k_{\mathbf{X}}(\mathbf{x},\mathbf{x})$ and $k_{\mathbf{Y}}(\mathbf{y},\mathbf{y})$ respectively, and they decrease as the new data point $\mathbf{x}$ gets closer to the training data $\mathbf{X}$. The uncertainty parameter $\phi(\mathbf{x},\mathbf{y})$ of SMTGP is given by $\phi(\mathbf{x},\mathbf{y}) = \frac{\eta_{\mathbf{x}}^{1-\theta}\eta_{\mathbf{y}}^{\theta}}{\eta_{\mathbf{xy}}}$, where $\eta_{\mathbf{xy}} = k^{\theta}_{\mathbf{XY}} - \mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY}}\mathbf{K}^{-1}_{\mathbf{XY}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY}}$, and it does not depend on the γ parameter. It was shown in [10] that the structured prediction with SMTGP maximizes this parameter resulting in a probabilistic interpretation of the SMTGP output and has a negative correlation with the error in structured prediction; this insight is missing in the original KLTGP framework.

B. IMPORTANCE WEIGHTED SMTGP FOR COVARIATE SHIFT

Under covariate shift, the GP regression model is given by (11), which is equivalent to re-weighting each input and output point by $w^{\frac{1}{2}}(\mathbf{x}_i^{\text{tr}})$ [29], [30].

$$w^{\frac{1}{2}}(\mathbf{x})y^{(d)} = w^{\frac{1}{2}}(\mathbf{x})f^{(d)}(\mathbf{x}) + \varepsilon^{(d)} \quad (11)$$

Thus, the weighted kernels are given by (12) where $\mathbf{W} = \text{diag}\{w(\mathbf{x}_1^{\text{tr}}), \ldots, w(\mathbf{x}_{n_{\text{tr}}}^{\text{tr}})\}$ is a diagonal matrix consisting of the importance weights $\mathbf{w}$ obtained by the Frank-Wolfe covariate shift algorithms proposed in Section III.

$$\mathbf{K_{X_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{K_X}\mathbf{W}^{\frac{1}{2}}, \quad \mathbf{k}^{\mathbf{x}}_{\mathbf{X_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{k}^{\mathbf{x}}_{\mathbf{X}}$$
$$\mathbf{K_{Y_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{K_Y}\mathbf{W}^{\frac{1}{2}}, \quad \mathbf{k}^{\mathbf{y}}_{\mathbf{Y_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{k}^{\mathbf{y}}_{\mathbf{Y}} \quad (12)$$

Further, in the importance weighted SMTGP (IW-SMTGP) framework, we have:

$$\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY_W}} = (1-\theta)\mathbf{k}^{\mathbf{x}}_{\mathbf{X_W}} + \theta\mathbf{k}^{\mathbf{y}}_{\mathbf{Y_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY}}$$
$$\mathbf{K_{XY_W}} = (1-\theta)\mathbf{K_{X_W}} + \theta\mathbf{K_{Y_W}} = \mathbf{W}^{\frac{1}{2}}\mathbf{K_{XY}}\mathbf{W}^{\frac{1}{2}} \quad (13)$$

The optimization function of SMTGP is modified to handle the covariate shift with weighted kernels and the corresponding IW-SMTGP predicted output $\hat{\mathbf{y}}_{\mathbf{w}}$ is given by (14).

$$\hat{\mathbf{y}}_{\mathbf{w}} = \underset{\mathbf{y}}{\text{argmin}}\ \frac{1}{\gamma - 1}\left[\left(k_{\mathbf{Y}}(\mathbf{y},\mathbf{y}) - \mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y_W}}^{-1}\mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}}\right)^{\frac{\theta(1-\gamma)}{2(1-\theta)}} \times \left(k^{\theta}_{\mathbf{XY}} - \mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY_W}}\mathbf{K}^{-1}_{\mathbf{XY_W}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY_W}}\right)^{\frac{-(1-\gamma)}{2(1-\theta)}}\right] \quad (14)$$

Using chain rule and re-arrangement, the derivative of the optimization function for IW-SMTGP with respect to the d^{th} dimension of structured output is given by (15). For the specific choice of Gaussian-RBF kernel, $\frac{\partial k_{\mathbf{Y}}(\mathbf{y},\mathbf{y})}{\partial y^{(d)}} = 0$ and $\frac{\partial \mathbf{k}^{\mathbf{y}}_{\mathbf{Y}}}{\partial y^{(d)}}$ is given by (16). Since $\mathbf{W}$ is a diagonal matrix, computing its inverse $\mathbf{W}^{-1}$ is trivial and thus, the computational complexity of IW-SMTGP is same as that of SMTGP.

$$\begin{aligned}\frac{\partial L_{\theta,\gamma}}{\partial y^{(d)}} = \frac{-\theta}{2(1-\theta)}\Bigg[&\left(k_{\mathbf{Y}}(\mathbf{y},\mathbf{y}) - \mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y_W}}^{-1}\mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}}\right)^{\frac{\theta(1-\gamma)}{2(1-\theta)}-1} \\ &\times \left(-2\mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y_W}}^{-1}\mathbf{W}^{\frac{1}{2}}\frac{\partial \mathbf{k}^{\mathbf{y}}_{\mathbf{Y}}}{\partial y^{(d)}}\right) \\ &\times \left(k^{\theta}_{\mathbf{XY}} - \mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY_W}}\mathbf{K}^{-1}_{\mathbf{XY_W}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY_W}}\right)^{\frac{-(1-\gamma)}{2(1-\theta)}}\Bigg] \\ &+ \frac{1}{2(1-\theta)}\Bigg[\left(k_{\mathbf{Y}}(\mathbf{y},\mathbf{y}) - \mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y_W}}^{-1}\mathbf{k}_{\mathbf{Y_W}}^{\mathbf{y}}\right)^{\frac{\theta(1-\gamma)}{2(1-\theta)}} \\ &\times \left(k^{\theta}_{\mathbf{XY}} - \mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY_W}}\mathbf{K}^{-1}_{\mathbf{XY_W}}\mathbf{k}^{\mathbf{xy}}_{\mathbf{XY_W}}\right)^{\frac{-(1-\gamma)}{2(1-\theta)}-1} \\ &\times \left(-2\mathbf{k}^{\mathbf{xy}\intercal}_{\mathbf{XY_W}}\mathbf{K}^{-1}_{\mathbf{XY_W}}\theta\mathbf{W}^{\frac{1}{2}}\frac{\partial \mathbf{k}^{\mathbf{y}}_{\mathbf{Y}}}{\partial y^{(d)}}\right)\Bigg] \end{aligned} \quad (15)$$

$$\frac{\partial \mathbf{k}^{\mathbf{y}}_{\mathbf{Y}}}{\partial y^{(d)}} = \begin{bmatrix} \frac{-1}{\rho_y^2}(y_1^{(d)\text{tr}} - y^d)k_{\mathbf{Y}}(\mathbf{y}_1,\mathbf{y}) \\ \frac{-1}{\rho_y^2}(y_2^{(d)\text{tr}} - y^d)k_{\mathbf{Y}}(\mathbf{y}_2,\mathbf{y}) \\ \vdots \\ \frac{-1}{\rho_y^2}(y_N^{(d)\text{tr}} - y^d)k_{\mathbf{Y}}(\mathbf{y}_N,\mathbf{y}) \end{bmatrix} \quad (16)$$

The uncertainty parameter of SMTGP is also updated to include the importance weights under covariate shift. Using weighted kernels, the importance weighted uncertainty

$\phi_{\mathbf{W}}(\mathbf{x}, \mathbf{y})$ of IW-SMTGP is derived as given by (17).

$$\begin{aligned}\phi_{\mathbf{W}}(\mathbf{x}, \mathbf{y}) &= \frac{\eta_{\mathbf{x}_\mathbf{W}}^{1-\theta}\eta_{\mathbf{y}_\mathbf{W}}^{\theta}}{\eta_{\mathbf{x}\mathbf{y}_\mathbf{W}}} \\ \eta_{\mathbf{x}_\mathbf{W}} &= k_{\mathbf{X}}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{X}_\mathbf{W}}^{\mathbf{x}\intercal}\mathbf{K}_{\mathbf{X}_\mathbf{W}}^{-1}\mathbf{k}_{\mathbf{X}_\mathbf{W}}^{\mathbf{x}} \\ \eta_{\mathbf{y}_\mathbf{W}} &= k_{\mathbf{Y}}(\mathbf{y}, \mathbf{y}) - \mathbf{k}_{\mathbf{Y}_\mathbf{W}}^{\mathbf{y}\intercal}\mathbf{K}_{\mathbf{Y}_\mathbf{W}}^{-1}\mathbf{k}_{\mathbf{Y}_\mathbf{W}}^{\mathbf{y}} \\ \eta_{\mathbf{x}\mathbf{y}_\mathbf{W}} &= k_{\mathbf{X}\mathbf{Y}}^{\theta} - \mathbf{k}_{\mathbf{X}\mathbf{Y}_\mathbf{W}}^{\mathbf{x}\mathbf{y}\intercal}\mathbf{K}_{\mathbf{X}\mathbf{Y}_\mathbf{W}}^{-1}\mathbf{k}_{\mathbf{X}\mathbf{Y}_\mathbf{W}}^{\mathbf{x}\mathbf{y}} \end{aligned} \tag{17}$$

Note that $\phi_{\mathbf{W}}(\mathbf{x}, \mathbf{y}) \leq \frac{|(1-\theta)\mathbf{K}_{\mathbf{X}_\mathbf{W}}+\theta\mathbf{K}_{\mathbf{Y}_\mathbf{W}}|}{|\mathbf{K}_{\mathbf{X}_\mathbf{W}}|^{1-\theta}|\mathbf{K}_{\mathbf{Y}_\mathbf{W}}|^{\theta}}$ due to the following inequality as an agreement between the joint distributions $p(\mathbf{X}, \mathbf{x})$ and $p(\mathbf{Y}, \mathbf{y})$:

$$\frac{|\mathbf{K}_{\mathbf{X}_\mathbf{W}}|^{1-\theta}|\mathbf{K}_{\mathbf{Y}_\mathbf{W}}|^{\theta}\eta_{\mathbf{x}_\mathbf{W}}^{1-\theta}\eta_{\mathbf{y}_\mathbf{W}}^{\theta}}{|(1-\theta)\mathbf{K}_{\mathbf{X}_\mathbf{W}} + \theta\mathbf{K}_{\mathbf{Y}_\mathbf{W}}|\eta_{\mathbf{x}\mathbf{y}_\mathbf{W}}} \leq 1$$

Equality is achieved only when the two joint distributions are identical, which justifies maximizing $\phi_{\mathbf{W}}(\mathbf{x}, \mathbf{y})$.

V. EXPERIMENTAL RESULTS

The proposed Frank-Wolfe covariate shift algorithms are applied to two benchmark applications of structured regression: human pose estimation and music mood estimation. These two applications are chosen because they both exhibit structured output whose multiple dimensions are correlated with each other and this property can be nicely exploited by SMTGP and IW-SMTGP frameworks. Human pose estimation is a challenging and active research topic in computer vision whereas music mood estimation is actively pursued by the MIR (music information retrieval) community. Estimating multi-dimensional pose from a single RGB image is known to be an ill-posed problem, since multiple articulations of body limbs may result in the same projection of pose and the system needs to be invariant to various factors such as clothing texture and shape, skin color, background scenes, lighting, etc. Music mood prediction and recommendation is a popular service in several commercial apps (e.g. Musicovery, Spotify, Wynk, etc.) that suggests an automated playlist based on the current mood query of the user. Improving the mood estimation of a particular audio clip and enhanced recommendations are among the major focus areas of the digital music industry.

The performance for these applications is evaluated using the following methods: the conventional Gaussian Process regression (GPR) [28] (note that the prediction is obtained separately for each output dimension in GPR), KLTGP [22], its importance weighted variant IW-KLTGP [29], SMTGP [10] and the proposed IW-SMTGP. RuLSIF [12] is used in IW-KLTGP [29] to determine the importance weights (with biased estimates), but in this work, we use the proposed Frank-Wolfe covariate shift algorithms to learn the KLIEP importance weights for IW-KLTGP to provide a fair comparison with IW-SMTGP.

A. COVARIATE SHIFT IN HUMAN POSE ESTIMATION

The HumanEva dataset [31] consists of synchronized multi-view video and motion capture data with Histogram of Oriented Gradients (HoG) extracted features ($\mathbf{x} \in \mathbb{R}^{270}$) of three subjects ($S_1, S_2, S_3$) performing the following actions: boxing, gesturing, jogging, throwing/catching and walking. The output multi-dimensional pose ($\mathbf{y} \in \mathbb{R}^{60}$) is encoded by (20) 3D joint markers using *torsoDistal* as the root, captured via three color cameras (C_1, C_2, C_3), with a total of $9,630$ image-pose frames for each camera. The following five covariate shift scenarios [29] are considered for human pose estimation:

a) *Selection Bias* (C_1): All three subjects are selected for training and only one subject is selected during testing with camera 1 data being used.
b) *Selection Bias* (C_{1-3}): Similar to above but with all three camera data used ($3 \times 4,815 = 14,445$ training and testing frames).
c) *Subject Transfer* (C_1): The test subject is not included during the training phase.
d) *Motion Transfer* (C_{1-3}): The motions used for training are boxing, gesturing and jogging, whereas the motions used for testing are throwing/catching and walking.
e) *Camera Transfer*: Camera 1 data (C_1) is used for training and Camera 2 data (C_2) is used for testing.

TABLE 1. Comparison of methods for estimating weights of the HumanEva dataset.

Covariate Shift	Method	# non-sparse atoms of $\boldsymbol{\alpha}$	Time (s) to find $\mathbf{w}$
Subject Transfer (C_1)	KLIEP	2019 (64%)	27.1±0.24
	FWCS	1317 (42%)	14.8±0.32
Train: S_2, S_3; Test: S_1	AFWCS	690 (22%)	8.3±0.16
(3135 test samples)	PFWCS	**471** (**15%**)	**5.4**±0.21
Motion Transfer (C_{1-3})	KLIEP	2608 (72%)	29.3±0.19
	FWCS	1377 (38%)	16.4±0.42
Train: S_1, S_2, S_3; Test: S_2	AFWCS	942 (26%)	8.9±0.33
(3622 test samples)	PFWCS	**689** (**19%**)	**6.2**±0.27

Table 1 shows the comparison of methods used to estimate the weights $\mathbf{w}$ for two covariate shift scenarios. In both cases, the PFWCS algorithm results in the lowest number of non-sparse atoms of $\boldsymbol{\alpha}$. Importance weighted cross-validation (IWCV) [24] is used to determine the optimal σ parameter of (1). The training run-time in seconds is measured over 50 trials and the average values (with standard deviation) is noted for all four methods to estimate the importance weights $\mathbf{w}$. PFWCS algorithm is faster than other methods since, in every iteration, it updates only two atoms related to the towards and away directions as opposed to updating all atoms in the FWCS algorithm. For further experiments, the PFWCS algorithm is used to compute the weights for all covariate shift scenarios of the HumanEva dataset.

We use randomly chosen 50% disjoint training and testing sets. In cases of fewer training samples (e.g. Figure 3), we randomly sub-sample n_{tr} from the full training set and repeat the sub-sampling procedure 10 times to avoid the sampling bias and report the average resulting joint errors. The joint error for each pose (in mm) is measured as the average Euclidean

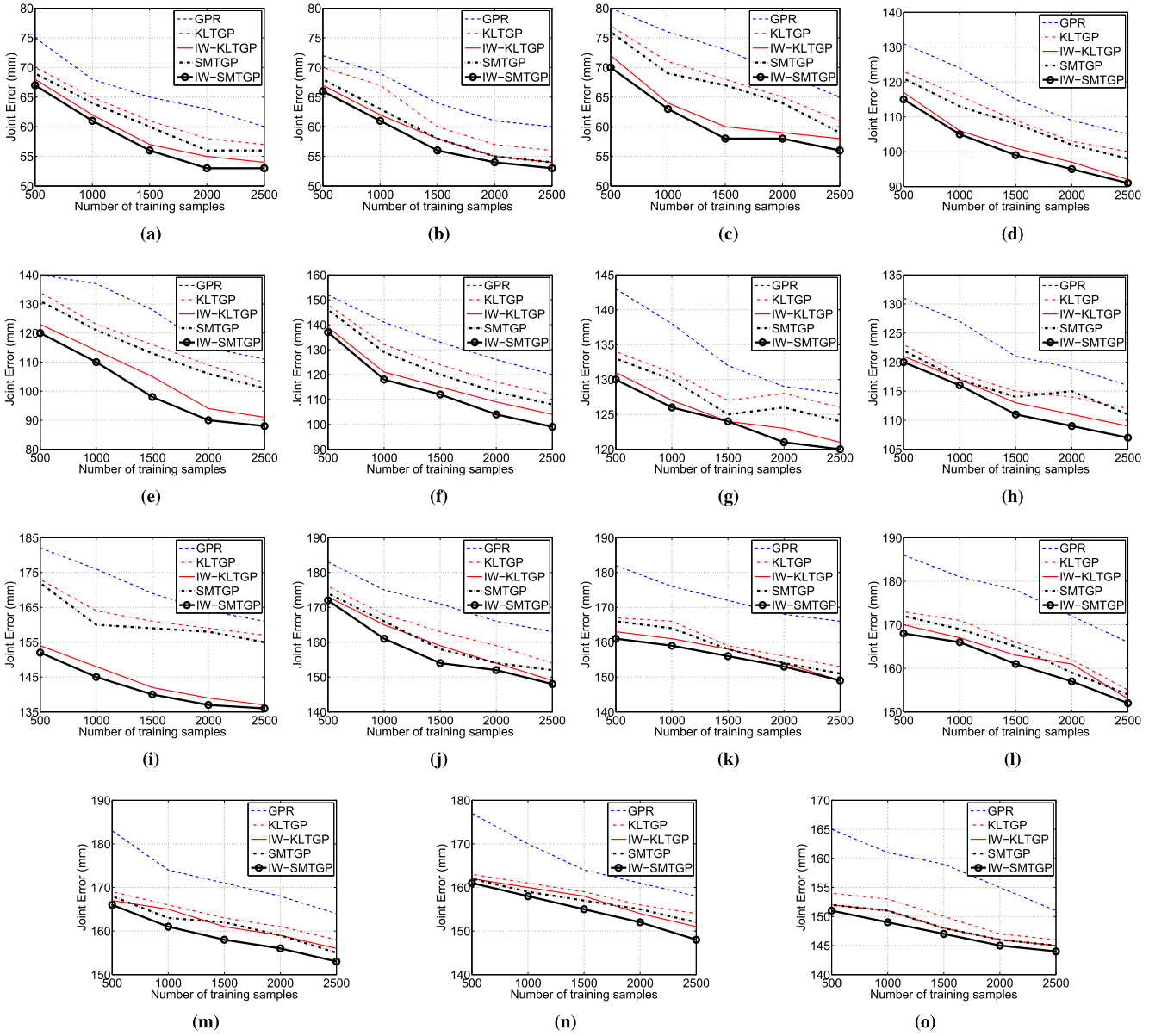

FIGURE 3. Performance on HumanEva dataset as a function of the number of training samples; results are averaged over all motions for each subject. The IW-SMTGP results are statistically significant as per *paired t-test* at 5% significance level. (a) Sel. Bias (C_1), S_1. (b) Sel. Bias (C_1), S_2. (c) Sel. Bias (C_1), S_3. (d) Sel. Bias (C_{1-3}), S_1 (e) Sel. Bias (C_{1-3}), S_2. (f) Sel. Bias (C_{1-3}), S_3. (g) Subject Transfer S_1. (h) Subject Transfer S_2. (i) Subject Transfer S_3. (j) Motion Transfer S_1. (k) Motion Transfer S_2. (l) Motion Transfer S_3. (m) Camera Transfer S_1, C_2. (n) Camera Transfer S_2, C_2. (o) Camera Transfer S_3, C_2.

distance given by (18), where $\hat{\mathbf{y}}$ is the estimated pose vector and $\mathbf{y}^*$ is the true pose vector. For TGPs, the original parameter setting of [22]: $\lambda_x = 10^{-3}, \lambda_y = 10^{-3}, 2\rho_x^2 = 5$ and $2\rho_y^2 = 5 \times 10^5$ is used. Further, for SMTGP, the model parameters are set to $\theta = 0.99$ and $\gamma = 0.99$ as stated in [10].

$$Error_{\text{pose}}(\hat{\mathbf{y}}, \mathbf{y}^*) = \frac{1}{20} \sum_{i=1}^{20} \|\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{*(i)}\| \tag{18}$$

Table 2 shows the performance of joint error as well as the uncertainties of SMTGP and IW-SMTGP using the full training set for various covariate shift scenarios. Both SMTGP and IW-SMTGP use the same parameter setting; however, the weights $\mathbf{w}$ learned by the PFWCS algorithm results in better estimation of the structured pose leading to reduced joint regression error with IW-SMTGP. Figure 3 compares the performance of baseline and existing methods and shows the average pose prediction error as a function of the number of training samples (averaged over all motions and 10 runs). The graphs show that the importance weighted methods improve the performance relative to the non-weighted counterparts. Also, IW-SMTGP results in a statistically significant performance using a *paired t-test* at 5% significance level relative to other methods.

TABLE 2. Performance on the HumanEva dataset averaged across all motions.

Covariate Shift	Train	Test	SMTGP Error ($\phi(\mathbf{x},\mathbf{y})$)	IW-SMTGP Error ($\phi_{\mathbf{W}}(\mathbf{x},\mathbf{y})$)
Selection Bias (C_1)	S_1, S_2, S_3	S_1	52.819 (0.992)	**51.679** (0.995)
		S_2	51.436 (0.993)	**50.273** (0.991)
		S_3	57.232 (0.983)	**55.302** (0.995)
Selection Bias (C_{1-3})	S_1, S_2, S_3	S_1	83.241 (0.974)	**81.788** (0.995)
		S_2	84.718 (0.980)	**82.828** (0.995)
		S_3	87.873 (0.991)	**85.912** (0.986)
Subject Transfer (C_1)	S_2, S_3	S_1	123.865 (0.978)	**119.772** (0.982)
	S_1, S_3	S_2	107.209 (0.986)	**104.353** (0.984)
	S_1, S_2	S_3	148.259 (0.989)	**134.719** (0.992)
Motion Transfer (C_{1-3})	S_1, S_2, S_3	S_1	144.896 (0.962)	**142.398** (0.976)
		S_2	154.717 (0.967)	**151.606** (0.981)
		S_3	148.308 (0.974)	**147.287** (0.983)
Camera Transfer (C_1)	S_1, S_2, S_3	S_1, C_2	149.709 (0.976)	**148.539** (0.982)
		S_2, C_2	148.435 (0.987)	**146.302** (0.992)
		S_3, C_2	145.499 (0.979)	**143.509** (0.964)

B. COVARIATE SHIFT IN MUSIC MOOD ESTIMATION

For music mood estimation, two benchmark datasets are used: AMG (All Music Guide) [32] and DEAM (Database for Emotion Analysis of Music) [33]. The AMG (DEAM) dataset contains crowdsourced annotations of 1608 (1802) audio clips of duration 30 seconds (45 seconds) from a variety of Western popular music genres (Rock, Pop, Electronic, etc.) annotated by 665 (195) users, along 2D valence and arousal (VA) dimensions with values in the range $[-1, 1]$. Since not all annotators annotated every audio clip and they may not be equally reliable, truth discovery analysis [34] of this crowdsourced data is required to determine the estimated consensus. Instead of taking the average VA value for each audio clip from the crowdsourced annotated data, the EM algorithm proposed in [35] is used to consider the reliability of each annotator and compute the estimated consensus ground truth $\mathbf{y} \in \mathbb{R}^2$. This estimated consensus algorithm also tackles the long-tail phenomenon commonly observed in crowdsourced data by using the upper bound of the confidence interval of χ^2 distribution.

For each audio clip, standard acoustic features are extracted using MIRToolbox [36] across four categories (dynamics, spectral, timbral and tonal) resulting in a 70-dimensional feature vector per frame of 50 ms duration with 50% overlap. For an effective prototypical feature representation, the variational Bayesian inference algorithm is used to compute the Bayesian Acoustic Gaussian Mixture Model (BAGMM) posterior probability feature vector $\mathbf{x} \in \mathbb{R}^{K_{opt}}$ for each audio clip, where $K_{opt} = 117$ [35]. The advantage of Bayesian inference is that the number of latent audio topics K can be determined from the data automatically, thus avoiding the problems of singularity and over/under-fitting with ad-hoc values of K [9].

The t-SNE (t-distributed Stochastic Neighbor Embedding) [37] scatterplot of these two datasets projected onto three dimensions is shown in Figure 4, where it can be observed that

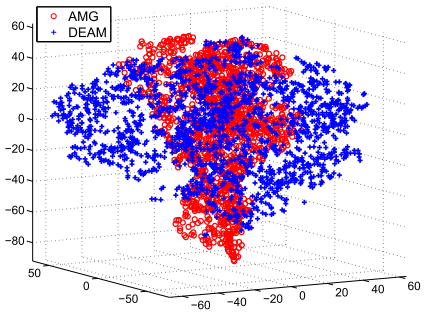

FIGURE 4. t-SNE scatterplot of music mood datasets.

the distributions of AMG and DEAM projected features are different from each other with some overlap between the two. Thus, covariate shift in the form of subject transfer [38] can be observed, since the two music mood datasets were developed independently with different selections of audio clips from various sources. The following two scenarios for music mood estimation are evaluated:

a) *Subject Transfer*: AMG dataset is used for the training phase and DEAM dataset is used for the testing phase.
b) *Subject Transfer*: DEAM dataset is used for the training phase and AMG dataset is used for the testing phase.

The comparison of methods for estimating weights $\mathbf{w}$ using importance weighted cross-validation for these two covariate shift scenarios is given in Table 3. The PFWCS algorithm results in sparser $\boldsymbol{\alpha}$ and is faster than other methods, where the average run-time in seconds (with standard deviation) over 50 trials is noted. The regression performance for music mood estimation is measured using three criteria: the coefficient of determination R^2 (higher is better), root mean square error *RMSE* (lower is better) given by (19) and the uncertainty parameter ($\phi(\mathbf{x},\mathbf{y})$ or $\phi_{\mathbf{W}}(\mathbf{x},\mathbf{y})$) individually for the valence and arousal dimensions. Here, $\hat{y}$ is the predicted mood dimension, y^* is the estimated consensus obtained from the crowdsourced data and $\bar{y}$ is the corresponding average value. For TGPs, the parameters were cross-validated with a grid search over suitable ranges [10], [22] and the parameters resulting in highest R^2 were obtained empirically: $\lambda_x = 10^{-4}$, $\lambda_y = 10^{-4}$, $2\rho_x^2 = 100$, $2\rho_y^2 = 1$, $\theta = 0.7$ and $\gamma = 0.99$.

$$R^2 = 1 - \frac{\sum_i (\hat{y}^{(i)} - y^{*(i)})^2}{\sum_i (\hat{y}^{(i)} - \bar{y})^2}$$
$$RMSE = \sqrt{\frac{1}{n_{\text{te}}} \sum_i (\hat{y}^{(i)} - y^{*(i)})^2} \tag{19}$$

Table 4 shows the performance of SMTGP and IW-SMTGP using all audio data samples from both datasets under the two scenarios of covariate shift. We observe the significance of importance weights learned by the PFWCS algorithm due to which IW-SMTGP outperforms SMTGP with the same parameter settings. However, valence estimation is still a challenging problem for music mood estimation.

TABLE 3. Comparison of methods for estimating weights of music mood datasets.

Covariate Shift	Method	# non-sparse atoms of $\boldsymbol{\alpha}$	Time (s) to find $\mathbf{w}$
Subject Transfer	KLIEP	1473 (82%)	18.2±0.14
	FWCS	629 (35%)	7.2±0.25
Train: AMG; Test: DEAM	AFWCS	415 (23%)	5.6±0.19
(1802 test samples)	PFWCS	**253** (**14%**)	**3.8**±0.16
Subject Transfer	KLIEP	1254 (78%)	16.4±0.10
	FWCS	530 (33%)	6.8±0.31
Train: DEAM; Test: AMG	AFWCS	338 (21%)	5.2±0.28
(1608 test samples)	PFWCS	**193** (**12%**)	**3.5**±0.11

TABLE 4. Performance evaluation for music mood estimation.

Train	Test	Mood dimension	SMTGP R^2 $RMSE, \phi(\mathbf{x},\mathbf{y})$	IW-SMTGP R^2 $RMSE, \phi_{\mathbf{W}}(\mathbf{x},\mathbf{y})$
AMG	DEAM	Arousal	0.721 0.169, 0.991	**0.732** **0.153**, 0.990
		Valence	0.512 0.236, 0.974	**0.526** **0.221**, 0.977
DEAM	AMG	Arousal	0.724 0.174, 0.987	**0.735** **0.162**, 0.989
		Valence	0.516 0.241, 0.972	**0.531** **0.228**, 0.975

Figure 5 shows the average R^2 values as a function of the number of training samples using GPR, KLTGP, IW-KLTGP, SMTGP, and the proposed IW-SMTGP. In this case too, subsampling of the training set is done 10 times to avoid the sampling bias and the average results are reported. It can be observed that SMTGP performs better than KLTGP and GPR, and IW-SMTGP results in a statistically significant performance using a *paired t-test* at 5% significance level relative to other methods.

VI. CONCLUSION

In this work, we proposed computationally efficient Away-steps Frank-Wolfe and Pairwise Frank-Wolfe Covariate Shift algorithms to correct the covariate shift of an importance weight estimation procedure (KLIEP) in an unsupervised manner resulting in sparser solutions. Due to the use of linear optimization in these algorithms, the time complexity per iteration is significantly smaller than the original projected gradient method of KLIEP. The proposed AFWCS and PFWCS algorithms perform better than the FWCS algorithm for determining the weights of the training data. The PFWCS algorithm achieves the most sparse solution with high computational efficiency. The convergence rate analysis showed that all proposed algorithms achieve a linear convergence rate. We also modified the Sharma-Mittal Twin Gaussian Process structured regression framework to handle the covariate shift and derived its importance weighted formulation having quadratic computational complexity (similar to KLTGP). Experimental evaluation validated the performance of proposed work on two applications of structured regression using benchmark datasets. In both cases of human pose and music

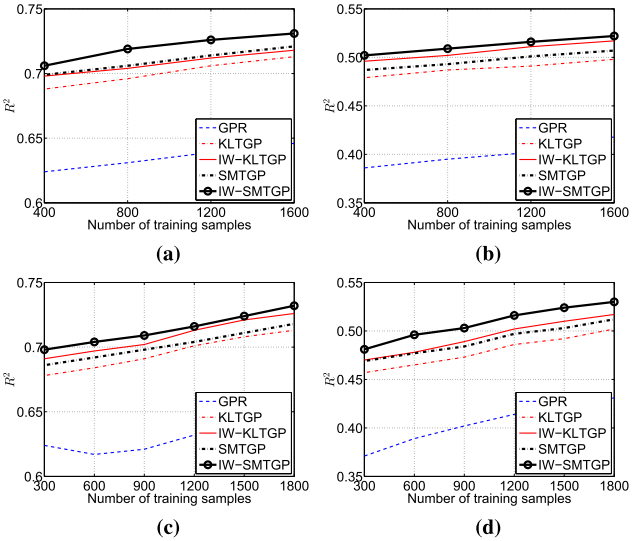

FIGURE 5. Performance on music mood datasets as a function of the number of training samples. The IW-SMTGP results are statistically significant as per *paired t-test* at 5% significance level. (a) Arousal (Train: AMG, Test: DEAM), (b) Valence (Train: AMG, Test: DEAM), (c) Arousal (Train: DEAM, Test: AMG), (d) Valence (Train: DEAM, Test: AMG).

mood estimation, the proposed approaches outperform the state-of-the-art techniques to eliminate the bias occurring due to covariate shift in various scenarios. For future work, it will be interesting to explore Frank-Wolfe optimization principles for covariate shift techniques in structured classification applications.

APPENDIX

A. ARMIJO LINE SEARCH CONDITION

The Armijo condition stated in Theorem 1 ensures that the objective function increases by a significant amount at each iteration t using inexact line search.

Theorem 1: Suppose F is a continuous and differentiable function, its gradient $\mathbf{g}_t$ is Lipschitz continuous with Lipschitz constant C, $\tau \in (0, 1)$ and $\mathbf{d}_t$ is the towards direction at the current solution $\boldsymbol{\alpha}_t$. Then, the Armijo condition given by (20) is satisfied for all $\rho \in [0, \rho^*]$, where $\rho^* = \frac{2(\tau-1)\langle \mathbf{g}_t, \mathbf{d}_t\rangle}{C\|\mathbf{d}_t\|_2^2}$.

$$F(\boldsymbol{\alpha}_t + \rho\mathbf{d}_t) \geq F(\boldsymbol{\alpha}_t) + \tau\rho\langle \mathbf{g}_t, \mathbf{d}_t\rangle \tag{20}$$

Proof: Using Taylor's series, we have

$$\begin{aligned} &F(\boldsymbol{\alpha}_t + \rho\mathbf{d}_t) \\ &\geq F(\boldsymbol{\alpha}_t) + \rho\langle \mathbf{g}_t, \mathbf{d}_t\rangle + \frac{1}{2}C\rho^2\|\mathbf{d}_t\|_2^2 \\ &\geq F(\boldsymbol{\alpha}_t) + \rho\langle \mathbf{g}_t, \mathbf{d}_t\rangle + \frac{1}{2}C\rho\frac{2(\tau-1)\langle \mathbf{g}_t, \mathbf{d}_t\rangle}{C\|\mathbf{d}_t\|_2^2}\|\mathbf{d}_t\|_2^2 \\ &= F(\boldsymbol{\alpha}_t) + \rho\langle \mathbf{g}_t, \mathbf{d}_t\rangle + \rho(\tau-1)\langle \mathbf{g}_t, \mathbf{d}_t\rangle \\ &= F(\boldsymbol{\alpha}_t) + \tau\rho\langle \mathbf{g}_t, \mathbf{d}_t\rangle \end{aligned}$$

■

B. DERIVATION OF UPDATES FOR FW ALGORITHMS

The update equations of atoms $\boldsymbol{\mu}_{t+1}$ and active set $\mathcal{S}_{t+1}$ for FWCS, AFWCS, and PFWCS algorithms are derived here

and the maximum step-size $\rho_{\max}$ is determined to guarantee feasibility of the solution.

1) FWCS ALGORITHM

For any ρ_t, we have

$$\begin{aligned}
&\boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\mathrm{FW}} \\
&= \boldsymbol{\alpha}_t + \rho_t(\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\alpha}_t) \\
&= (1-\rho_t)\boldsymbol{\alpha}_t + \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} = (1-\rho_t)\sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&= \sum_{l \neq l_t^{\mathrm{FW}}} (1-\rho_t)\boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + (1-\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{FW}})\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} + \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&= \sum_{l \neq l_t^{\mathrm{FW}}} (1-\rho_t)\boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \{(1-\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{FW}}) + \rho_t\}\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&= \sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{FW}}(l)\boldsymbol{\beta}^{(l)} \qquad (21)
\end{aligned}$$

To ensure feasibility of the solution, we need $\sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{FW}}(l) = 1$ and $\hat{\boldsymbol{\mu}}_t^{\mathrm{FW}}(l) \geq 0$. From (21), it follows that

$$\begin{aligned}
\sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{FW}}(l) &= \sum_{l \neq l_t^{\mathrm{FW}}} (1-\rho_t)\boldsymbol{\mu}_t(l) + (1-\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{FW}}) + \rho_t \\
&= (1-\rho_t)\sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l) + \rho_t = 1. \because \sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l) = 1
\end{aligned}$$

For $l = 1, \ldots, n_{\mathrm{te}}$, $\hat{\boldsymbol{\mu}}_t^{\mathrm{FW}}(l) \geq 0 \;\; \forall \rho_t \in [0, 1]$, thus $\rho_{\max} = 1$. The update equations of $\boldsymbol{\mu}_{t+1}$ and $\mathcal{S}_{t+1}$ for FWCS algorithm are thus given by (22 – 23).

$$\boldsymbol{\mu}_{t+1}(l) = \begin{cases} (1-\rho_t)\boldsymbol{\mu}_t(l) & \text{if } l \neq l_t^{\mathrm{FW}}, \\ (1-\rho_t)\boldsymbol{\mu}_t(l) + \rho_t & \text{if } l = l_t^{\mathrm{FW}}. \end{cases} \qquad (22)$$

$$\mathcal{S}_{t+1} = \begin{cases} \{l_t^{\mathrm{FW}}\} & \text{if } \rho_t = 1, \\ \mathcal{S}_t & \text{if } \rho_t < 1 \;\&\; l_t^{\mathrm{FW}} \in \mathcal{S}_t, \\ \mathcal{S}_t \cup \{l_t^{\mathrm{FW}}\} & \text{if } \rho_t < 1 \;\&\; l_t^{\mathrm{FW}} \in \mathcal{S}_t. \end{cases} \qquad (23)$$

2) AFWCS ALGORITHM

If the chosen direction is $\mathbf{d}_t = \mathbf{d}_t^{\mathrm{FW}}$ (towards), then $\rho_{\max} = 1$. But if $\mathbf{d}_t = \mathbf{d}_t^{\mathrm{AFW}}$ (away), then $\rho_{\max} = 1$ may produce an infeasible solution. For any ρ_t, we have

$$\begin{aligned}
&\boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\mathrm{AFW}} \\
&= \boldsymbol{\alpha}_t + \rho_t(\boldsymbol{\alpha}_t - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}) \\
&= (1+\rho_t)\boldsymbol{\alpha}_t - \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&= (1+\rho_t)\sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} - \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&= \sum_{l \neq l_t^{\mathrm{AFW}}} (1+\rho_t)\boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + (1+\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})\boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} \\
&\quad - \rho_t \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} \\
&= \sum_{l \neq l_t^{\mathrm{AFW}}} (1+\rho_t)\boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \{(1+\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) - \rho_t\}\boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} \\
&= \sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l)\boldsymbol{\beta}^{(l)} \qquad (24)
\end{aligned}$$

To ensure feasibility of the solution, we need $\sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l) = 1$ and $\hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l) \geq 0$. From (24), it follows that

$$\begin{aligned}
\sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l) &= \sum_{l \neq l_t^{\mathrm{AFW}}} (1+\rho_t)\boldsymbol{\mu}_t(l) + (1+\rho_t)\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) - \rho_t \\
&= (1+\rho_t)\sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l) - \rho_t = 1. \because \sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l) = 1
\end{aligned}$$

For $l = 1, \ldots, n_{\mathrm{te}}$, $\hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l) \geq 0 \;\; \forall \rho_t \in [0, 1]$, and $\hat{\boldsymbol{\mu}}_t^{\mathrm{AFW}}(l_t^{\mathrm{AFW}}) \geq 0$ for $0 \leq \rho_t \leq \rho_{\max}$, where the maximum step-size $\rho_{\max}$ is given by (25).

$$(1+\rho_{\max})\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) - \rho_{\max} = 0 \therefore \rho_{\max} = \frac{\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})}{1 - \boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})} \qquad (25)$$

Note that $\rho_{\max}$ is well-defined since $\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})$ cannot be 1. Suppose that $\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) = 1$, then it implies $\boldsymbol{\alpha}_t = \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}$, i.e. $\mathbf{d}_t^{\mathrm{AFW}} = 0$, thus $\langle \mathbf{g}_t, \mathbf{d}_t^{\mathrm{FW}} \rangle \geq 0 = \langle \mathbf{g}_t, \mathbf{d}_t^{\mathrm{AFW}} \rangle$, which is a contradiction as it will lead to a choice of the towards step. When $\rho_t = \rho_{\max}$ and $\rho_{\max} < 1$ (or equivalently, $\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) < \frac{1}{2}$), this condition is known as a *drop step* since it cannot guarantee a decrease of the duality gap. The location l_t^{AFW} is removed from the active set $\mathcal{S}_t$ for the *drop step*. The update equations of $\boldsymbol{\mu}_{t+1}$ and $\mathcal{S}_{t+1}$ for AFWCS algorithm are thus given by (26 – 27).

$$\boldsymbol{\mu}_{t+1}(l) = \begin{cases} (1+\rho_t)\boldsymbol{\mu}_t(l) & \text{if } l \neq l_t^{\mathrm{AFW}}, \\ (1+\rho_t)\boldsymbol{\mu}_t(l) - \rho_t & \text{if } l = l_t^{\mathrm{AFW}}. \end{cases} \qquad (26)$$

$$\mathcal{S}_{t+1} = \begin{cases} \mathcal{S}_t \setminus \{l_t^{\mathrm{AFW}}\} & \text{if } \rho_t = \rho_{\max}(\textit{drop step}), \\ \mathcal{S}_t & \text{if } \rho_t < \rho_{\max} \;\&\; l_t^{\mathrm{FW}} \in \mathcal{S}_t, \\ \mathcal{S}_t \cup \{l_t^{\mathrm{FW}}\} & \text{if } \rho_t < \rho_{\max} \;\&\; l_t^{\mathrm{FW}} \notin \mathcal{S}_t. \end{cases} \qquad (27)$$

3) PFWCS ALGORITHM

For any ρ_t, we have

$$\begin{aligned}
&\boldsymbol{\alpha}_t + \rho_t \mathbf{d}_t^{\mathrm{PFW}} \\
&= \boldsymbol{\alpha}_t + \rho_t(\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}) \\
&= \sum_{l=1}^{n_{\mathrm{te}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \rho_t(\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}) \\
&= \sum_{l \neq l_t^{\mathrm{FW}}, l \neq l_t^{\mathrm{AFW}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \boldsymbol{\mu}_t(l_t^{\mathrm{FW}})\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} + \boldsymbol{\mu}_t(l_t^{\mathrm{AFW}})\boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} \\
&\quad + \rho_t(\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} - \boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})}) \\
&= \sum_{l \neq l_t^{\mathrm{FW}}, l \neq l_t^{\mathrm{AFW}}} \boldsymbol{\mu}_t(l)\boldsymbol{\beta}^{(l)} + \{\boldsymbol{\mu}_t(l_t^{\mathrm{FW}}) + \rho_t\}\boldsymbol{\beta}^{(l_t^{\mathrm{FW}})} \\
&\quad + \{\boldsymbol{\mu}_t(l_t^{\mathrm{AFW}}) - \rho_t\}\boldsymbol{\beta}^{(l_t^{\mathrm{AFW}})} \\
&= \sum_{l=1}^{n_{\mathrm{te}}} \hat{\boldsymbol{\mu}}_t^{\mathrm{PFW}}(l)\boldsymbol{\beta}^{(l)} \qquad (28)
\end{aligned}$$

To ensure feasibility of the solution, we need $\sum_{l=1}^{n_{\text{te}}} \hat{\boldsymbol{\mu}}_t^{\text{PFW}}(l) = 1$ and $\hat{\boldsymbol{\mu}}_t^{\text{PFW}}(l) \geq 0$. From (28), it follows that

$$\begin{aligned}
&\sum_{l=1}^{n_{\text{te}}} \hat{\boldsymbol{\mu}}_t^{\text{PFW}}(l) \\
&\quad = \sum_{\substack{l \neq l_t^{\text{FW}} \\ l \neq l_t^{\text{AFW}}}} \boldsymbol{\mu}_t(l) + \{\boldsymbol{\mu}_t(l_t^{\text{FW}}) + \rho_t\} + \{\boldsymbol{\mu}_t(l_t^{\text{AFW}}) - \rho_t\} \\
&\quad = \sum_{l=1}^{n_{\text{te}}} \boldsymbol{\mu}_t(l) + \rho_t - \rho_t = 1. \because \sum_{l=1}^{n_{\text{te}}} \boldsymbol{\mu}_t(l) = 1
\end{aligned}$$

For $l = 1, \ldots, n_{\text{te}}$, $\hat{\boldsymbol{\mu}}_t^{\text{PFW}}(l) \geq 0 \ \forall \rho_t \in [0, 1]$, and $\hat{\boldsymbol{\mu}}_t^{\text{PFW}}(l_t^{\text{AFW}}) \geq 0$ for $0 \leq \rho_t \leq \rho_{\max}$, where the maximum step-size $\rho_{\max}$ is given by (29). The update equations of $\boldsymbol{\mu}_{t+1}$ and $\mathcal{S}_{t+1}$ for PFWCS algorithm are thus given by (30 – 31).

$$\boldsymbol{\mu}_t(l_t^{\text{AFW}}) - \rho_{\max} = 0 \therefore \rho_{\max} = \boldsymbol{\mu}_t(l_t^{\text{AFW}}) \tag{29}$$

$$\boldsymbol{\mu}_{t+1}(l) = \begin{cases} \boldsymbol{\mu}_t(l) & \text{if } l \neq l_t^{\text{FW}} \& l \neq l_t^{\text{AFW}}, \\ \boldsymbol{\mu}_t(l) + \rho_t & \text{if } l = l_t^{\text{FW}}, \\ \boldsymbol{\mu}_t(l) - \rho_t & \text{if } l = l_t^{\text{AFW}}. \end{cases} \tag{30}$$

$$\mathcal{S}_{t+1} = \begin{cases} \mathcal{S}_t \setminus \{l_t^{\text{AFW}}\} & \text{if } \rho_t = \rho_{\max} \ \& \ l_t^{\text{FW}} \in \mathcal{S}_t, \\ & (\textit{drop step}) \\ \mathcal{S}_t \cup \{l_t^{\text{FW}}\} \setminus \{l_t^{\text{AFW}}\} & \text{if } \rho_t = \rho_{\max} \ \& \ l_t^{\text{FW}} \notin \mathcal{S}_t, \\ & (\textit{swap step}) \\ \mathcal{S}_t & \text{if } \rho_t < \rho_{\max} \ \& \ l_t^{\text{FW}} \in \mathcal{S}_t, \\ \mathcal{S}_t \cup \{l_t^{\text{FW}}\} & \text{if } \rho_t < \rho_{\max} \ \& \ l_t^{\text{FW}} \notin \mathcal{S}_t. \end{cases} \tag{31}$$

C. JOINT KERNEL DECOMPOSITION

Using Aitken block-diagonalization formula [39], the joint kernel matrix $\mathbf{K}_{\mathbf{X}\cup\mathbf{x}}$ can be decomposed as (32), since it is square and non-singular. Here, $\eta_{\mathbf{x}} = k_{\mathbf{X}}(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{\mathbf{X}}^{\mathbf{x}\intercal} \mathbf{K}_{\mathbf{X}}^{-1} \mathbf{k}_{\mathbf{X}}^{\mathbf{x}}$ is the Schur complement of $\mathbf{K}_{\mathbf{X}\cup\mathbf{x}}$ with respect to $\mathbf{K}_{\mathbf{X}}$. Thus, the determinant of the joint kernel matrix is given by $|\mathbf{K}_{\mathbf{X}\cup\mathbf{x}}| = |\mathbf{K}_{\mathbf{X}}| \times \eta_{\mathbf{x}}$, since $\eta_{\mathbf{x}}$ is a scalar and the determinant of unit triangular matrices is one. Similarly, $|\mathbf{K}_{\mathbf{Y}\cup\mathbf{y}}| = |\mathbf{K}_{\mathbf{Y}}| \times \eta_{\mathbf{y}}$, where $\eta_{\mathbf{y}} = k_{\mathbf{Y}}(\mathbf{y}, \mathbf{y}) - \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}\intercal} \mathbf{K}_{\mathbf{Y}}^{-1} \mathbf{k}_{\mathbf{Y}}^{\mathbf{y}}$ is the Schur complement of the joint kernel matrix $\mathbf{K}_{\mathbf{Y}\cup\mathbf{y}}$ with respect to $\mathbf{K}_{\mathbf{Y}}$.

$$\mathbf{K}_{\mathbf{X}\cup\mathbf{x}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{k}_{\mathbf{X}}^{\mathbf{x}\intercal} \mathbf{K}_{\mathbf{X}}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{\mathbf{X}} & \mathbf{0} \\ \mathbf{0} & \eta_{\mathbf{x}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{K}_{\mathbf{X}}^{-1} \mathbf{k}_{\mathbf{X}}^{\mathbf{x}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{32}$$

ACKNOWLEDGMENT

We thank the following authors: Liefeng Bo and Cristian Sminchisescu for making the features of HumanEva dataset publicly available, Mohamed Elhoseiny for sharing the SMTGP implementation and Yi-Hsuan Yang for sharing the audio clips of AMG dataset via personal communication.

REFERENCES

[1] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, Jan. 2012.

[2] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Ann. Inst. Stat. Math.*, vol. 60, no. 4, pp. 699–746, Aug. 2008.

[3] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, vol. 28, no. 1, Jun. 2013, pp. 427–435.

[4] A. Joulin, K. Tang, and L. Fei-Fei, "Efficient image and video co-localization with Frank-Wolfe algorithm," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 8694. Cham, Switzerland: Springer, Sep. 2014, pp. 253–268.

[5] J. Wen, R. Greiner, and D. Schuurmans, "Correcting covariate shift with the Frank-Wolfe algorithm," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, Jul. 2015, pp. 1010–1016.

[6] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "VNect: Real-time 3D human pose estimation with a single RGB camera," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–44, Jul. 2017.

[7] S. Kinauer, R. Güler, S. Chandra, and I. Kokkinos, "Structured output prediction and learning for deep monocular 3D human pose estimation," in *Proc. 11th Int. Conf. Energy Minimization Methods Comput. Vis. Pattern Recognit. (EMMCVPR)*, Venice, Italy, Oct. 2017, pp. 1–14.

[8] S. Fukayama and M. Goto, "Music emotion recognition with adaptive aggregation of Gaussian process regressors," in *Proc. 41st IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 71–75.

[9] J. C. Wang, Y. H. Yang, H. M. Wang, and S. K. Jeng, "Modeling the affective content of music with a Gaussian mixture model," *IEEE Trans. Affective Computing*, vol. 6, no. 1, pp. 56–68, Jan. 2015.

[10] M. Elhoseiny and A. Elgammal, "Generalized twin Gaussian processes using Sharma–Mittal divergence," *Mach. Learn.*, vol. 100, nos. 2–3, pp. 399–424, Sep. 2015.

[11] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *J. Mach. Learn. Res.*, vol. 10, pp. 1391–1445, Jul. 2009.

[12] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, "Relative density-ratio estimation for robust distribution comparison," *Neural Comput.*, vol. 25, no. 5, pp. 1324–1370, May 2013.

[13] M. Yamada and M. Sugiyama, "Direct importance estimation with Gaussian mixture models," *IEICE Trans. Inf. Syst.*, vol. 92, no. 10, pp. 2159–2162, Oct. 2009.

[14] D. Garber and E. Hazan, "Faster rates for the Frank-Wolfe method over strongly-convex sets," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 541–549.

[15] S. Lacoste-Julien and M. Jaggi, "On the global linear convergence of Frank-Wolfe optimization variants," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Montréal, QC, Canada, Dec. 2015, pp. 496–504.

[16] J. GuéLat and P. Marcotte, "Some comments on Wolfe's 'away step,'" *Math. Program.*, vol. 35, no. 1, pp. 110–119, May 1986.

[17] A. Beck and S. Shtern, "Linearly convergent away-step conditional gradient for non-strongly convex functions," *Math. Program.*, vol. 164, nos. 1–2, pp. 1–27, Jul. 2017.

[18] S. Reddi, S. Sra, B. Póczos, and A. Smola, "Stochastic Frank-Wolfe methods for nonconvex optimization," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 2016, pp. 1244–1251.

[19] A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. K. Dokania, and S. Lacoste-Julien, "Minding the gaps for block Frank-Wolfe optimization of structured SVMs," in *Proc. 33rd Int. Conf. Mach. Learn. (PMLR)*, New York, NY, USA, vol. 48, Jun. 2016, pp. 593–602.

[20] T. Baltrušaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.

[21] G. Sandbach, S. Zafeiriou, and M. Pantic, "Markov random field structures for facial action unit intensity estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Sydney, NSW, Australia, Dec. 2013, pp. 738–745.

[22] L. Bo and C. Sminchisescu, "Twin Gaussian processes for structured prediction," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, p. 28, Mar. 2010.

[23] F. Nielsen and R. Nock, "A closed-form expression for the Sharma–Mittal entropy of exponential families," *J. Phys. A, Math. Theor.*, vol. 45, no. 3, Dec. 2011, Art. no. 032003.
[24] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, May 2007.
[25] W. Sun and Y. Ya-Xiang, *Optimization Theory and Methods: Nonlinear Programming*. New York, NY, USA: Springer-Verlag, 2006.
[26] H. Allende, E. Frandi, R. Ñanculef, and C. Sartori, "Pairwise away steps for the Frank-Wolfe algorithm," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, CA, USA, Dec. 2013, pp. 1–5.
[27] R. Ñanculef, E. Frandi, C. Sartori, and H. Allende, "A novel Frank-Wolfe algorithm: Analysis and applications to large-scale SVM training," *Inf. Sci.*, vol. 285, pp. 66–99, Nov. 2014.
[28] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[29] M. Yamada, L. Sigal, and M. Raptis, "Covariate shift adaptation for discriminative 3D pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 235–247, Feb. 2014.
[30] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Planning Inference*, vol. 90, no. 2, pp. 227–244, 2000.
[31] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, pp. 4–27, 2010.
[32] Y.-A. Chen, Y.-H. Yang, J.-C. Wang, and H. Chen, "The AMG1608 dataset for music emotion recognition," in *Proc. 40th IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 693–697.
[33] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLoS ONE*, vol. 12, no. 3, Mar. 2017, Art. no. e0173392.
[34] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, Apr. 2010.
[35] S. Chapaneri and D. Jayaswal, "Structured prediction of music mood with twin Gaussian processes," in *Pattern Recognition and Machine Intelligence* (Lecture Notes in Computer Science), vol. 10597. Cham, Switzerland: Springer, 2017, pp. 647–654.
[36] O. Lartillot and P. Toiviainen, "A MATLAB toolbox for musical feature extraction from audio," in *Proc. Int. Conf. Digit. Audio Effects (DAFx)*, Sep. 2017, pp. 237–244.
[37] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[38] C. Cortes and M. Mohri, "Domain adaptation in regression," in *Proc. 22nd Int. Conf. Algorithmic Learn. Theory (ALT)*, in Lecture Notes in Computer Science, vol. 6925. Berlin, Germany: Springer, Oct. 2011, pp. 308–323.
[39] Y. Tian and Y. Takane, "More on generalized inverses of partitioned matrices with Banachiewicz–Schur forms," *Linear Algebra Appl.*, vol. 430, nos. 5–6, pp. 1641–1655, Mar. 2009.

SANTOSH V. CHAPANERI (M'11) received the B.E. degree in electronics and telecommunication engineering from the University of Mumbai, India, in 2001, and the M.S. degree in electrical and computer engineering from the University of Arizona, USA, in 2008. He was a Software Developer with Patni Computer Systems Ltd., Mumbai, India, and Microsoft Corporation, Seattle, USA. He is currently an Assistant Professor with the St. Francis Institute of Technology, University of Mumbai. His research interests include machine learning and signal processing. He is a Reviewer of *IET Communications*, *IET Electronics Letters*, *IET Signal Processing*, and *IET Transactions on Image Processing*.

DEEPAK J. JAYASWAL received the B.E. degree in electronics engineering from Shivaji University, India, in 1991, the M.Tech. degree in communication engineering from IIT Bombay, India, in 2002, and the Ph.D. degree in computer engineering from the National Institute of Technology, Surat, India, in 2010. He is currently a Professor with the St. Francis Institute of Technology, University of Mumbai, India. His research interests include image and video processing and machine learning problems. He is a Reviewer of the *IETE Journal of Education*, *Evolutionary Intelligence* (Springer), and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.