No Bias Left Behind: Covariate Shift Adaptation for Discriminative 3D Pose Estimation

Makoto Yamada[1], Leonid Sigal[2], Michalis Raptis[2]

[1]NTT Communication Science Laboratories
[2]Disney Research Pittsburgh
yamada@cs.titech.ac.jp, {lsigal, mraptis}@disneyresearch.com

Abstract. Discriminative, or (structured) prediction, methods have proved effective for variety of problems in computer vision; a notable example is 3D monocular pose estimation. All methods to date, however, relied on an assumption that training (source) and test (target) data come from the same underlying joint distribution. In many real cases, including standard datasets, this assumption is flawed. In presence of training set bias, the learning results in a biased model whose performance degrades on the (target) test set. Under the assumption of covariate shift we propose an unsupervised domain adaptation approach to address this problem. The approach takes the form of training instance re-weighting, where the weights are assigned based on the ratio of training and test marginals evaluated at the samples. Learning with the resulting *weighted* training samples, alleviates the bias in the learned models. We show the efficacy of our approach by proposing weighted variants of Kernel Regression (KR) and Twin Gaussian Processes (TGP). We show that our weighted variants outperform their un-weighted counterparts and improve on the state-of-the-art performance in the public (HumanEva) dataset.

1 Introduction

Many problems in computer vision can be expressed in the form of (structured) predictions of real-valued multivariate output, $\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{y}}}$, from a high-dimensional multivariate input, $\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{x}}}$. In this paper, we focus on such models in the context of articulated 3D pose estimation.

Articulated 3D pose estimation, particularly from monocular images and/or video, is a challenging problem due to variability in person appearance, pose, body shape, lighting, and motion. Despite these challenges, discriminative methods, have proved to be effective in recovering the 3D pose [1–17] in variety of scenarios. In these methods the goal is to learn a direct (and often multimodal) mapping, $f : \mathbb{R}^{d_{\mathrm{x}}} \rightarrow \mathbb{R}^{d_{\mathrm{y}}}$, from image features (e.g., bag-of-words of HoG or SIFT descriptors) $\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{x}}}$ to 3D poses $\boldsymbol{y} \in \mathbb{R}^{d_{\mathrm{y}}}$, typically expressed as joint positions or angles. Probabilistic formulations do so by learning the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ based on the training dataset of n_{tr} image-pose pairs – $\{(\boldsymbol{x}_i^{\mathrm{tr}}, \boldsymbol{y}_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ (assumed to be independent and identically distributed (i.i.d.) samples from the underlying joint density $p_{\mathrm{tr}}(\boldsymbol{x}, \boldsymbol{y})$). A number of methods [1–17] of this form have been proposed over the last decade that explore a gamut

of models, image features and learning architectures. However, in *all* cases it has been assumed that the training and test distributions are one and the same, i.e., $p_{\text{tr}}(\boldsymbol{x}, \boldsymbol{y}) = p_{\text{te}}(\boldsymbol{x}, \boldsymbol{y})$, and hence the model learned using the training feature-pose pairs can be directly applied to the test image features, $\boldsymbol{x}^{\text{te}}$, to infer the output 3D pose $\boldsymbol{y}^{\text{te}}$.

The problem of *dataset bias* is starting to emerge as very prominent issue in object categorization [18–22], where even large datasets (e.g., LabelMe or ImageNet) have shown to exhibit significant (and often unexpected) biases [22] in the form of lighting, object appearance and viewpoint to name a few. We argue that similar issues exist in 3D pose estimation and need to be addressed if one is to build a system that works outside of well calibrated laboratory setups and datasets. The issues of dataset bias and overfitting to the training set, in 3D pose estimation, are evident from poor generalization that one often sees when applying such models to novel data. In addition, we argue that *within* dataset bias is, at least in certain cases, as prevalent as the *between* dataset bias. While in dataset creation an effort is typically made to make the training and test sets as similar as possible, this is difficult to achieve precisely. For example, Urtasun and Darrell in [16] show that performance decreases dramatically (to as low as 25% of the baseline) when training and test sequences are disjoint[1]. Note that this is despite the fact that, even in the disjoint case, training and test sequences were captured by the same static cameras and with no appreciable difference in subject appearance and lighting. Similar degradation of performance is often observed when a subject is not included in the training set, or when training data comes from multiple subjects (and/or motions) and at the test time only a single subject (and/or motion) is observed [5].

Unfortunately, the domain adaptation approaches proposed in [18–21] are not adequate for addressing the bias in this case. First, they typically assume categorical classification, as opposed to multi-valued (structured) predictions. Second, and more importantly, they are supervised and assume existence of one or more labeled instances from the test set to allow the transfer learning to fine tune the source model to a target test set. In many scenarios, such as, 3D pose estimation, obtaining 3D pose for a test image is infeasible. To this end, we formulate a novel training instance re-weighting mechanism for addressing the bias in (structured) prediction problems under the assumption of a *covariate shift* [24] in an unsupervised manner; where we assume that $p(\boldsymbol{y}|\boldsymbol{x}) = p_{\text{tr}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\text{te}}(\boldsymbol{y}|\boldsymbol{x})$, but the marginals are different ($p_{\text{tr}}(\boldsymbol{x}) \neq p_{\text{te}}(\boldsymbol{x})$).

Contributions: The key contributions of this work is to shed some light on the potential issues of dataset bias in the structured prediction problems, mainly, 3D pose estimation, and to propose a simple, yet effective, solution for handling such bias through training instance re-weighting in a covariate shift adaptation formulation. We illustrate the efficacy of our approach by proposing weighted variants of Kernel Regression and Twin Gaussian Processes and showing that

[1] The baseline sampled training/test sets on per-frame bases from the full HumanEva-I [23] dataset.

they outperform their non-weighted counterparts in various setups and with different image features. As a consequence we achieve state-of-the-art performance on HumanEva-I dataset. The proposed training instance re-weighting, however, is general and is amenable to most popular formulations (e.g., Linear Regression, Mixture of Experts, GPLVM, Kernel Information Embeddings), as well as to other (structured) prediction problems in computer vision.

2 Related Work

Discriminative models are popular in vision for various tasks, including 3D human pose [1–17], human shape [25], hand pose [26, 27] and face pose [28] estimation. The focus of this paper is on 3D pose estimation which we discuss next; we also discuss transfer learning techniques that motivated our approach.

A variety of (structured) prediction methods have been proposed for 3D pose estimation in the literature, including Nearest Neighbor regression (NN) [13], linear Locally-Weighted Regression (LWR) [13], Linear Regression (LR) [2], Relevance Vector Regression (RVR) [2], Kernel Regression (KR) [13] and Gaussian Process Regression (GPR) [17]. The observation that the mapping from image features to 3D pose is typically multi-modal, due to inherent imaging ambiguities, has led to introduction of multi-modal alternatives and mixture models, including Mixture of Linear Regressors (MoLR) [1], conditional Bayesian Mixture of Experts (cMoE) [4, 8, 14, 15], Local GP Regression (LGPR) [16] and Twin Gaussian Processes (TGP) [5], to name a few. Mixture models, such as MoLR and cMoE, can produce multiple solutions (one for each expert) with the hope that ambiguities can be resolved by an oracle [4, 8] or over time [14]; alternatively, optimization can be used to ascend to the most prominent mode of the conditional distribution [5]. We leverage these prior methods and propose an Importance Weighted Twin Gaussian Processes (IWTGP) model, based on TGP [5], where importance weights adopt the model to the test data at hand in an un-supervised fashion.

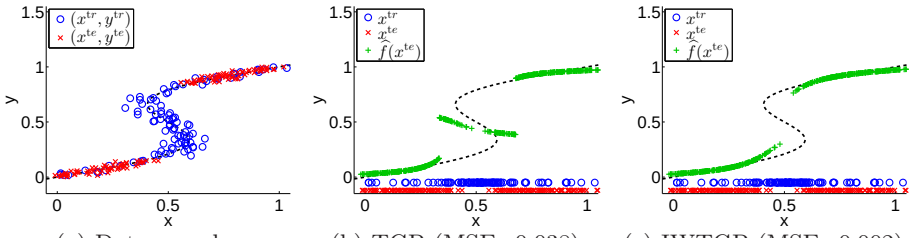
The methods outlined above differ significantly in learning and inference. The issue of learning from large datasets was addressed in [4] using a forward feature selection and bound optimization, allowing training of cMoE models from upward of 100,000 input-output samples. A competing issue of learning from small datasets has also received much attention, with most methods converging on intermediate shared low-dimensional latent representations (e.g., shared GPLVM (sGPLVM) [6, 9] or shared Kernel Information Embeddings (sKIE) [12]) to address overfitting with few input-output samples; some formulations were shown to be amenable semi-supervised learning settings [8, 9, 12] where a large number of *unpaired* marginal samples, which are drawn from the training distribution (not test distribution), are available. We deal with training from large datasets, as in [16] and [5], by first selecting an active set of input-output pairs (k Nearest Neighbors to the test input feature vector $\boldsymbol{x}$) and then learning an IWTGP model for this reduced set. This results in a fixed model and inference complexity regardless of training set size (apart from the initial kNN lookup).

Our method is also motivated by recent works that study effects of dataset biases in vision. The issue of dataset bias has recently emerged as a serious problem in object categorization, with Torralba and Efros [22] showing that significant biases exist in all current datasets. As a result, techniques for *domain adaptation* in object categorization are starting to emerge [18–21]. However, unlike our method, the focus of such techniques, so far, has been on a supervised setting where one or more labeled examples are available at test time (in the target domain). This allows the source models, obtained using training data, to be adopted to the target test domain explicitly. A more recent variant by Kulis *et al.* [19] introduces a method for doing this in a cross-domain setting, where the representation of the features at train and test time may in itself be different. Our setting, here, is substantially different, however, as we assume that no labeled instances are present at test time. This makes the problem more challenging, but at the same time more realistic for our target application, as it is unreasonable to assume that accurate 3D pose can be annotated for monocular test images. This setting is a special case of domain adaptation known as *covariate shift* [24], where the training distribution $p_{\mathrm{tr}}(\boldsymbol{x})$ and test distribution $p_{\mathrm{te}}(\boldsymbol{x})$ over the inputs are different (i.e., $p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$) but the conditional distribution of output values, $p(\boldsymbol{y}|\boldsymbol{x})$, remains same.

The influence of the covariate shift could be mitigated by re-weighting of the log likelihood terms according to their importance within the test set. Since the importance is generally unknown, the key issue of covariate shift adaptation is to estimate these importance weights accurately. Following this idea, several direct importance weight estimation methods have been recently proposed [29–32]. In this paper, we adopt a novel importance weight estimation method called *relative unconstrained least-squares importance fitting* (RuLSIF) [32], since it holds practical advantage over competing methods. Mainly, it is computationally efficient and can naturally control the *adaptiveness* to the test distribution. In contrast to [32], however, we adopt RuLSIF for (structured) real-valued predictions and illustrate it's efficacy on a real-world vision problem.

3 Covariate Shift in 3D Pose Estimation

At first glance, it may not be evident why dataset bias plays a role in discriminative models, considering that discriminative methods are trying to model the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ and it seems reasonable to assume that $p(\boldsymbol{y}|\boldsymbol{x}) = p_{\mathrm{tr}}(\boldsymbol{y}|\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{y}|\boldsymbol{x})$ (even if $p_{\mathrm{tr}}(\boldsymbol{x}, \boldsymbol{y}) \neq p_{\mathrm{te}}(\boldsymbol{x}, \boldsymbol{y})$). In other words, how can the fact that $p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$ effect the conditional distribution? The issue is that the conditional models assume a certain functional form and typically choose the optimal parameters (within this functional form) by minimizing the average regression error (i.e., average discrepancy between the predicted and true values on the *training* set). Intuitively this means that the learned model performs more accurately in the denser regions than in the sparser regions of $p_{\mathrm{tr}}(\boldsymbol{x})$, because the denser regions dominate the average regression error. Hence, if $p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$, the learned model may no longer be *optimal* for the test set.



(a) Data samples (b) TGP (MSE=0.038) (c) IWTGP (MSE=0.002)

Fig. 1. Predicted outputs y by TGP (b) and IWTGP (c) under covariate shift in green. (a) Samples from the model $x = y + 0.3\sin(2\pi y) + e$ where $e \sim \mathcal{N}(0, 0.05^2)$; $\circ$ and $\times$ are training and test samples respectively (for clarity we also illustrate marginals $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$ in (b) and (c) bottom). Note that the input-output test samples are not used in the training of TGP and the output test samples are not used in the training of IWTGP, they are plotted in the figure for illustration purposes.

The formulation outlined in the previous paragraph is known in the transfer learning community as one of *covariate shift* [24]. Under covariate shift setup it is assumed that labeled training image-pose pairs $\{(\boldsymbol{x}_i^{\mathrm{tr}}, \boldsymbol{y}_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ drawn i.i.d. from $p(\boldsymbol{y}|\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})$ and unlabeled test image features $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ drawn i.i.d. from $p_{\mathrm{te}}(\boldsymbol{x})$ (which is usually different from $p_{\mathrm{tr}}(\boldsymbol{x})$) are available. The goal of (structured) prediction is to learn a mapping, $\boldsymbol{f} : \mathbb{R}^{d_{\mathrm{x}}} \to \mathbb{R}^{d_{\mathrm{y}}}$, which in the most general form can be expressed as:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{e} \in \mathbb{R}^{d_{\mathrm{y}}}$ is the noise. Under covariate shift this mapping is learned based on a *weighted* set of training image-pose pairs $\{(w_i, \boldsymbol{x}_i^{\mathrm{tr}}, \boldsymbol{y}_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$. Re-weighting each training instances by the ratio (a.k.a., importance weight), $w_i = w_1(\boldsymbol{x}_i^{\mathrm{tr}}; \boldsymbol{\theta}) = \frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})}$, removes the training set bias producing an unbiased model under assumption of covariate shift [24]. Note $\{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are necessary to estimate the numerator. The main challenge, however, is estimation of the importance weight; we discuss this in detail in Section 4.

Before proceeding, however, we would like to illustrate the effect of *covariate shift* on a synthetic toy example. In Figure 1, we illustrate the efficacy of our re-weighting scheme under covariate shift by incorporating it into Twin Gaussian Process (TGP); for details see Section 5.2. As can be seen, Importance Weighted TGP (IWTGP) can predict the true test output well, while standard TGP fails to predict the true test output, in particular, around $x = 0.5$. Note that in this specific case the mean squared error (MSE) is improved by a large margin (from 0.038 to 0.002) by incorporating the importance weight into the learning.

4 Importance Weight Estimation

The importance weight may be computed by separately estimating densities $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$ from training and test feature vectors and then taking their

ratio. However, density estimation is known to be a hard problem and taking the ratio of estimated densities tends to increase the estimation error [30]. Thus, this two step approach is not appropriate in practice. We adopt a method that allows us to directly learn the importance weight function without going through density estimation. The method is called the *relative unconstrained least-squares importance fitting* (RuLSIF) [32].

Let us first define the *relative importance weight* [32]:

$$w_\alpha(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{(1-\alpha)p_{\rm te}(\boldsymbol{x}) + \alpha p_{\rm tr}(\boldsymbol{x})}, \ 0 \leq \alpha \leq 1, \tag{2}$$

where α is the tuning parameter to control the *adaptiveness* to the test distribution. If $\alpha = 0$ (i.e., $w_0(\boldsymbol{x}) = 1$) gives no adaptation, while $\alpha = 1$ (i.e., $w_1(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{p_{\rm tr}(\boldsymbol{x})}$) gives the full adaptation from $p_{\rm tr}(\boldsymbol{x})$ to $p_{\rm te}(\boldsymbol{x})$; $0 < \alpha < 1$ will give an intermediate estimator[2].

Let $\mathcal{X}^{\rm tr}(\subseteq \mathbb{R}^{d_{\rm x}})$ be the domain of training image feature vector $\boldsymbol{x}^{\rm tr}$ and $\mathcal{X}^{\rm te}(\subseteq \mathbb{R}^{d_{\rm y}})$ be the domain of test image feature vector $\boldsymbol{x}^{\rm te}$. Suppose we are given $n_{\rm tr}$ and $n_{\rm te}$ i.i.d. training and test image feature vectors, $\{\boldsymbol{x}_i^{\rm tr} \mid \boldsymbol{x}_i^{\rm tr} \in \mathcal{X}^{\rm tr}, \ i = 1, \ldots, n_{\rm tr}\}$, $\{\boldsymbol{x}_j^{\rm te} \mid \boldsymbol{x}_j^{\rm te} \in \mathcal{X}^{\rm te}, \ j = 1, \ldots, n_{\rm te}\}$, drawn from distributions with densities $p_{\rm tr}(\boldsymbol{x})$ and $p_{\rm te}(\boldsymbol{x})$, respectively.

The final goal of relative importance weight estimation is to estimate the relative importance weight based on the training and test image features. Let us model the relative importance weight $w_\alpha(\boldsymbol{x})$ by the following kernel model:

$$w_\alpha(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\ell=1}^{n_{\rm te}} \theta_\ell \ \kappa(\boldsymbol{x}, \boldsymbol{x}_\ell^{\rm te}) = \sum_{\ell=1}^{n_{\rm te}} \theta_\ell \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_\ell^{\rm te}\|^2}{2\tau^2}\right), \tag{3}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n_{\rm te}})^\top$ are parameters to be learned from data samples, $^\top$ denotes the transpose, $\kappa(\cdot, \cdot)$ is the Gaussian kernel and τ (> 0) is the kernel bandwidth.

The parameters $\boldsymbol{\theta}$ in the model $w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})$ are determined so that the following expected squared-error J is minimized:

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{2}\mathbb{E}_{q_\alpha(\boldsymbol{x})}\left[\left(w_\alpha(\boldsymbol{x}; \boldsymbol{\theta}) - w_\alpha(\boldsymbol{x})\right)^2\right] \\ &= \frac{(1-\alpha)}{2}\mathbb{E}_{p_{\rm te}(\boldsymbol{x})}\left[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})^2\right] + \frac{\alpha}{2}\mathbb{E}_{p_{\rm tr}(\boldsymbol{x})}\left[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})^2\right] - \mathbb{E}_{p_{\rm te}(\boldsymbol{x})}[w_\alpha(\boldsymbol{x}; \boldsymbol{\theta})] + {\rm Const.}, \end{aligned}$$

where $q_\alpha(\boldsymbol{x}) = (1-\alpha)p_{\rm te}(\boldsymbol{x}) + \alpha p_{\rm tr}(\boldsymbol{x})$, and we used $w_\alpha(\boldsymbol{x})q_\alpha(\boldsymbol{x}) = p_{\rm te}(\boldsymbol{x})$ in the third term (see supplemental materials for derivation[3]).

[2] $\alpha = 1$ (i.e., $w_1(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{p_{\rm tr}(\boldsymbol{x})}$) gives the full adaptation from $p_{\rm tr}(\boldsymbol{x})$ to $p_{\rm te}(\boldsymbol{x})$. However, since the importance weight $w_1(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{p_{\rm tr}(\boldsymbol{x})}$ can diverge to infinity under a rather simple setting, e.g., when the ratio of two Gaussian function is considered [33], the estimation of $w_1(\boldsymbol{x}) = \frac{p_{\rm te}(\boldsymbol{x})}{p_{\rm tr}(\boldsymbol{x})}$ is unstable and the covariate shift adaptation tends to be unstable [24]. To cope with this instability issue, setting α to $0 < \alpha < 1$ is practically useful for stabilizing the covariate shift adaptation, even though it cannot give an unbiased model under covariate shift [32].

[3] `http://www.cs.brown.edu/~ls/Publications/eccv2012_supplemental.pdf`

Approximating the expectations by empirical averages, we obtain the following optimization problem:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{n_{\mathrm{te}}}}{\operatorname{argmin}} \left[\frac{1}{2} \boldsymbol{\theta}^\top \widehat{\boldsymbol{H}} \boldsymbol{\theta} - \widehat{\boldsymbol{h}}^\top \boldsymbol{\theta} + \frac{\nu}{2} \boldsymbol{\theta}^\top \boldsymbol{\theta} \right], \tag{4}$$

where $\nu \boldsymbol{\theta}^\top \boldsymbol{\theta}/2$ is included to avoid overfitting, and ν (≥ 0) denotes the regularization parameter. $\widehat{\boldsymbol{H}}$ is the $n_{\mathrm{te}} \times n_{\mathrm{te}}$ matrix with the (ℓ, ℓ')-th element

$$\widehat{H}_{\ell,\ell'} = \frac{(1-\alpha)}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \kappa(\boldsymbol{x}_i^{\mathrm{te}}, \boldsymbol{x}_\ell^{\mathrm{te}}) \kappa(\boldsymbol{x}_i^{\mathrm{te}}, \boldsymbol{x}_{\ell'}^{\mathrm{te}}) + \frac{\alpha}{n_{\mathrm{tr}}} \sum_{j=1}^{n_{\mathrm{tr}}} \kappa(\boldsymbol{x}_j^{\mathrm{tr}}, \boldsymbol{x}_\ell^{\mathrm{te}}) \kappa(\boldsymbol{x}_j^{\mathrm{tr}}, \boldsymbol{x}_{\ell'}^{\mathrm{te}});$$

$\widehat{\boldsymbol{h}}$ is the n_{te}-dimensional vector with the ℓ-th element $\widehat{h}_\ell = \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \kappa(\boldsymbol{x}_i^{\mathrm{te}}, \boldsymbol{x}_\ell^{\mathrm{te}})$. Then the solution to Eq. (4) can be *analytically* obtained as

$$\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{H}} + \nu \boldsymbol{I})^{-1} \widehat{\boldsymbol{h}}, \tag{5}$$

where $\boldsymbol{I}$ is the $n_{\mathrm{te}} \times n_{\mathrm{te}}$-dimensional identity matrix.

The performance of RuLSIF depends on the choice of the kernel bandwidth τ and the regularization parameter ν. Model selection of RuLSIF is possible based on cross-validation with respect to the squared-error criterion J [32].

Computational complexity: Learning RuLSIF has complexity $O(n_{\mathrm{te}}^3)$ due to the matrix inversion. However, when the number of test data is large, we may reduce the number of kernels in Eq.(3) to $b_{\mathrm{te}} (< n_{\mathrm{te}})$. Then, the inverse matrix in Eq.(5) can be efficiently computed with complexity $O(b_{\mathrm{te}}^3)$.

5 Importance Weighted 3D Human Pose Estimation

Given the derivation of the importance weight estimator, in previous section, we now formulate two regression-based methods that take these weights into account. We start by formulating Importance Weighted Kernel Regression (IWKR), which has a particularly simple form and allows learning of non-linear mapping between the image features and the 3D pose. IWKR, similar to standard KR, is well suited for unimodal predictions. However, in 3D pose estimation, the mapping from image features to 3D pose has been shown to be multi-modal, due to the inherent imaging ambiguities [14]. To address this, we also introduce an Importance Weighted Twin Gaussian Process model, based on [5], which in addition imposes structure on the output 3D poses. As a result, IWTGP is able to estimate the most prominent mode, corresponding to the most likely 3D pose, as opposed to averaging across modes as is the case with KR and IWKR.

5.1 Importance Weighted Kernel Regression

In kernel regression vector-valued regression function $\boldsymbol{f}$, in Eq.(1), takes the following form:

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{A}) = \boldsymbol{A}^\top \boldsymbol{k}(\boldsymbol{x}), \tag{6}$$

where $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{d_\mathrm{y}}] \in \mathbb{R}^{(n_\mathrm{tr}+1)\times d_\mathrm{y}}$ is a model parameter, d_y is the dimensionality of pose $\boldsymbol{y}$, $\boldsymbol{k}(\boldsymbol{x}) = [1, K(\boldsymbol{x}, \boldsymbol{x}_1^\mathrm{tr}), K(\boldsymbol{x}, \boldsymbol{x}_2^\mathrm{tr}), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_{n_\mathrm{tr}}^\mathrm{tr})]^\top$, and $K(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function. We use the Gaussian kernel [34] in our experiments: $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\rho_\mathrm{x}^2}\right)$, where ρ_x is the kernel bandwidth.

Under covariate shift setup, the use of *relative importance weighted* risk minimization was shown to be useful for adaptation from $p_\mathrm{tr}(\boldsymbol{x})$ to $p_\mathrm{te}(\boldsymbol{x})$ [32]:

$$\min_{\boldsymbol{A}} \left[\sum_{i=1}^{n_\mathrm{tr}} w_\alpha(\boldsymbol{x}_i^\mathrm{tr}) \|\boldsymbol{y}_i^\mathrm{tr} - \boldsymbol{A}^\top \boldsymbol{k}(\boldsymbol{x}_i^\mathrm{tr})\|^2 + \frac{\gamma}{2} \sum_{j=1}^{d_\mathrm{y}} \|\boldsymbol{\alpha}_j\|^2 \right], \tag{7}$$

where $w_\alpha(\boldsymbol{x})$ is the *relative importance weight* in Eq.(2), α is the tuning parameter to control the *adaptiveness* to the test distribution, and $\gamma \geq 0$ is the regularization parameter; we call this *importance weighted kernel regression* (IWKR).

The solution to Eq.(7) can be obtained analytically by

$$\widehat{\boldsymbol{A}} = (\widetilde{\boldsymbol{K}}^\mathrm{tr} \boldsymbol{W} (\widetilde{\boldsymbol{K}}^\mathrm{tr})^\top + \gamma \boldsymbol{I}) \widetilde{\boldsymbol{K}}^\mathrm{tr} \boldsymbol{W} (\boldsymbol{Y}^{tr})^\top, \tag{8}$$

where $\widetilde{\boldsymbol{K}}^\mathrm{tr} = [\boldsymbol{k}(\boldsymbol{x}_1^\mathrm{tr}), \ldots, \boldsymbol{k}(\boldsymbol{x}_{n_\mathrm{tr}}^\mathrm{tr})] \in \mathbb{R}^{(n_\mathrm{tr}+1)\times n_\mathrm{tr}}$, $\boldsymbol{Y}^{tr} = [\boldsymbol{y}_1^\mathrm{tr}, \ldots, \boldsymbol{y}_{n_\mathrm{tr}}^\mathrm{tr}] \in \mathbb{R}^{d_y \times n_\mathrm{tr}}$, and $\boldsymbol{W}$ is the $n_\mathrm{tr} \times n_\mathrm{tr}$-dimensional diagonal matrix with (i,i)-th diagonal element defined by $\boldsymbol{W}_{i,i} = w_\alpha(\boldsymbol{x}_i^\mathrm{tr})$.

The above IWKR method includes two tuning parameters: kernel parameter ρ and the regularization parameter γ. These parameters can be selected using importance-weighted variant of cross-validation (IWCV) [35].

Computational complexity: Learning IWKR has complexity $O((n_\mathrm{tr}+1)^3)$. Similar to RuLSIF, when the number of training data is large, we may reduce the number of kernels in Eq.(6) to $b_\mathrm{tr}(< n_\mathrm{tr}+1)$. Then, the inverse matrix in Eq.(8) can be efficiently computed with complexity $O(b_\mathrm{tr}^3)$. Since IWKR also includes the estimation of relative importance weight and its complexity is $O(b_\mathrm{te}^3)$. Thus, the complexity of IWKR is $O(b_\mathrm{tr}^3) + O(b_\mathrm{te}^3)$.

5.2 Importance Weighted Twin Gaussian Process

We now propose the importance-weighted variant of twin Gaussian processes [5] called IWTGP. The benefit of IWTGP over IWKR is that it can naturally take into account the multi-modality present in the human pose estimation, by incorporating structure over the output poses into the regression.

The Gaussian Process (GP) regression assumes a linear model in the function space with Gaussian noise for the k-th dimension (e.g., joint position):

$$y_k = f_k(\boldsymbol{x}) + e_k, \;\; e_k \sim \mathcal{N}(0, \sigma^2), \qquad f_k(\boldsymbol{x}) = \boldsymbol{\beta}_k^\top \boldsymbol{\phi}(\boldsymbol{x}), \tag{9}$$

where there is a zero mean Gaussian prior over the parameters $\boldsymbol{\beta}_k \sim \mathcal{N}(\boldsymbol{0}_p, \boldsymbol{\Sigma}_p)$; $\boldsymbol{0}_p$ is the p-dimensional zero vector and $\boldsymbol{\Sigma}_p$ is the p-dimensional covariance matrix, $\boldsymbol{\phi}(\boldsymbol{x})$ is the function which maps a d_x dimensional input vector $\boldsymbol{x}$ into

an p dimensional feature space. To make prediction for the test sample, one needs to average over all possible parameter values, weighted by their posterior, resulting in a Gaussian predictive distribution. GP has similar problems with multi-modality as KR. To address this limitation, TGP encodes the relations between both inputs and outputs using GP priors. This is achieved by minimizing the Kullback-Leibler divergence between the marginal GP of outputs (poses) and observations (features); we refer the reader to [5] for derivation.

As a result, the estimated pose in TGP is given as the solution of the following optimization problem [5]:

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\mathbb{R}^{d_{\mathrm{y}}}}{\operatorname{argmin}} \Big[L(\boldsymbol{y},\boldsymbol{y}) - 2\boldsymbol{l}(\boldsymbol{y})^{\top}\boldsymbol{u} - \eta \log\left[L(\boldsymbol{y},\boldsymbol{y}) - \boldsymbol{l}(\boldsymbol{y})^{\top}(\boldsymbol{L}+\lambda_{\mathrm{y}}\boldsymbol{I})^{-1}\boldsymbol{l}(\boldsymbol{y}) \right] \Big], \quad (10)$$

where $\boldsymbol{u} = (\boldsymbol{K} + \lambda_{\mathrm{x}}\boldsymbol{I})^{-1}\boldsymbol{k}(\boldsymbol{x})$, $\eta = K(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^{\top}\boldsymbol{u}$, $K(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\rho_{\mathrm{x}}^2}\right)$ and $L(\boldsymbol{y},\boldsymbol{y}') = \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{y}'\|^2}{2\rho_{\mathrm{y}}^2}\right)$ are the Gaussian kernel function for image feature vector $\boldsymbol{x}$ and pose feature vector $\boldsymbol{y}$, ρ_{x} and ρ_{y} are the kernel bandwidth, $\boldsymbol{l}(\boldsymbol{y}) = [L(\boldsymbol{y},\boldsymbol{y}_1),\ldots,L(\boldsymbol{y},\boldsymbol{y}_{n_{\mathrm{tr}}})]^{\top}$, $\boldsymbol{k}(\boldsymbol{x}) = [K(\boldsymbol{x},\boldsymbol{x}_1),\ldots,K(\boldsymbol{x},\boldsymbol{x}_{n_{\mathrm{tr}}})]^{\top}$, and λ_{y} and λ_{x} are regularization parameters to avoid overfitting. This optimization problem can be solved using a second order, BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection [5].

Under covariate shift, the likelihood of Gaussian Process can be given as [24]

$$\prod_{i=1}^{n_{\mathrm{tr}}} p(y_i^{tr}|\boldsymbol{x}_i^{\mathrm{tr}},\boldsymbol{\beta})^{w_\alpha(\boldsymbol{x}_i^{\mathrm{tr}})} \propto \prod_{i=1}^{n_{\mathrm{tr}}} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}})y_i^{tr} - w_\alpha^{\frac{1}{2}}(\boldsymbol{x}_i^{\mathrm{tr}})\boldsymbol{\phi}(\boldsymbol{x}_i^{\mathrm{tr}})^{\top}\boldsymbol{\beta}\|^2}{2\sigma^2} \right), \quad (11)$$

where $w_\alpha(\boldsymbol{x})$ is the relative importance weight function. Note, if we consider the MAP estimate for Eq. (11) with a prior distribution over $\boldsymbol{\beta}$, then we can show that IWKR and Eq. (11) are one and the same.

Thus, the GP regression model under covariate shift can be represented by

$$w_\alpha^{\frac{1}{2}}(\boldsymbol{x})y_k = w_\alpha^{\frac{1}{2}}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})^{\top}\boldsymbol{\beta}_k + e_k,\ e_k \sim \mathcal{N}(0,\sigma^2). \quad (12)$$

That is, to achieve covariate shift adaptation in TGP, we need to simply reweight each input and output by $w_\alpha^{\frac{1}{2}}(\boldsymbol{x})$. Therefore, the output of the importance weighted TGP (IWTGP) is given by

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\mathbb{R}^{d_{\mathrm{y}}}}{\operatorname{argmin}} \Big[L(\boldsymbol{y},\boldsymbol{y}) - 2\boldsymbol{l}(\boldsymbol{y})^{\top}\boldsymbol{u}_w - \eta_w \log\left[L(\boldsymbol{y},\boldsymbol{y}) - \boldsymbol{l}(\boldsymbol{y})^{\top}\boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{L}\boldsymbol{W}^{\frac{1}{2}}+\lambda_{\mathrm{y}}\boldsymbol{I})^{-1}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{l}(\boldsymbol{y}) \right] \Big], \quad (13)$$

where $\boldsymbol{u}_w = \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} + \lambda_{\mathrm{x}}\boldsymbol{I})^{-1}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{k}(\boldsymbol{x})$, $\eta_w = K(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^{\top}\boldsymbol{u}_w$. IWTGP can also be solved using a second order, BFGS quasi-Newton optimizer with cubic polynomial line search for optimal step size selection. We ignore the weighting for certain terms that are independent of $\boldsymbol{y}$, and hence do not effect the optimization, for simplicity.

Computational complexity: IWTGP requires matrix inversions of $n_{\mathrm{tr}} \times n_{\mathrm{tr}}$ matrices, the complexity of solving Eq.(13) is $O(n_{\mathrm{tr}}^3)$, which is impractical when n_{tr} is large. To deal with this issue, we first find the M nearest neighbors of a test input and estimate IWTGP on the reduced set of training paired samples. Then, the inverse matrix in Eq.(13) can be efficiently computed with complexity $O(M^3)$. IWTGP also includes the estimation of relative importance weight, thus the total complexity of IWTGP is $O(M^3) + O(b_{\mathrm{te}}^3)$.

5.3 Importance Weighting for Other Methods

The proposed weighting methodology is amenable to most popular formulations (e.g., Linear Regression, Mixture of Experts (MoE), GPLVM, KIE), as well as to other (structured) prediction problems in computer vision. For example, in Linear Regression importance weighting can be incorporate via Weighted Linear Regression. Incorporating importance weighting into MoE would amount to secondary weighting on top of expert assignment; for MoE models with soft expert assignments this would require very minor changes to the learning procedure. Latent variants like GPLVM and KIE can also make use of the importance weighting, for example, in KIE the importance weighted version of Mutual Information can be used to learn an IWKIE model.

6 Experiments

We compare the performance of the proposed methods IWKR and IWTGP with their un-weighted counterparts, KR [1] and TGP (we use public implementation from [5]), and weighted k-Nearest Neighbors approach (WkNN) [13]. We report performance on two publicly available datasets: Poser [2] and HumanEva-I [23].

Parameters: For Poser dataset, we experimentally (through grid search) set the TGP and IWTGP parameters to $\lambda_{\mathrm{x}} = \lambda_{\mathrm{y}} = 10^{-4}$, $2\rho_{\mathrm{x}}^2 = 5$, and $2\rho_{\mathrm{y}}^2 = 5000$. For HumanEva-I dataset, we used the original parameter setting of [5]: $\lambda_{\mathrm{x}} = \lambda_{\mathrm{y}} = 10^{-3}$, $2\rho_{\mathrm{x}}^2 = 5$, and $2\rho_{\mathrm{y}}^2 = 5 \times 10^5$. The number of M nearest neighbors in TGP and IWTGP is set to $min(800, n_{\mathrm{tr}})$. In RuLSIF, we set the $\alpha = 0.5$ and $b_{\mathrm{te}} = \min(500, n_{\mathrm{te}})$. For KR and IWKR, we set $b_{\mathrm{tr}} = \min(500, n_{\mathrm{tr}})$, and all the parameters are chosen by cross-validation (CV) and importance weighted CV; in WkNN we set the number of nearest neighbors to 25. In addition, instead of using the entire test set to adopt the model, we use a temporal window of 20 frames (feature vectors) around the current test sample to compute the importance weight for IWTGP and IWKR. This is more efficient and is also more realistic, as one will typically not see the full set of test examples all at once.

Computational speed: The overhead for importance weighting is small compared to the base methods; for example, IWTGP is about 4% slower than TGP when entire training set is used. Moreover, experimentally we observed that IWTGP can be faster than TGP with few samples (see supplemental materials). We attribute this to the fact that weighting in TGP can lead to an easier optimization problem, offsetting the coast of the weight estimation itself.

Table 1. Performance of IWKR and IWTGP on Poser dataset.

	IWTGP	TGP	**IWKR**	KR	NN [12]	GPLVM [6]	sKIE [12]
Error (deg)	**5.75**	5.83	**5.72**	6.04	6.87	6.50	5.77/5.95

6.1 Poser Dataset

Poser dataset [2] consists of 1927 training and 418 test images, which are synthetically generated, using Poser software package, from motion capture (Mocap) data (54 joint angles per frame). The image features, corresponding to bag-of-words representation with silhouette-based shape context features, and error metric are provided with the dataset [2]. Since the Poser data is synthetically generated and was tuned to unimodal predictions [2], there exists only a small bias between training and test images/features.

Error metric: The proposed error measure amounts to the root mean square error (in degrees), averaged over all joints angles, and is given by: $Error_{pose}(\widehat{\boldsymbol{y}}, \boldsymbol{y}^*) = \frac{1}{54}\sum_{m=1}^{54} \|(\widehat{\boldsymbol{y}}^{(m)} - \boldsymbol{y}^{*(m)}) \bmod 360^\circ\|$, where $\widehat{\boldsymbol{y}} \in \mathbb{R}^{54}$ is an estimated pose vector, and $\boldsymbol{y}^* \in \mathbb{R}^{54}$ is a true pose vector.

Performance: Table 1 shows the pose estimation result averaged across the test set. Proposed IWKR and IWTGP outperform their un-weighted counterparts, reducing error by 5% and 2% respectively. IWKR and IWTGP also compare favorably with other existing methods reported elsewhere. It is worth mentioning that Shared KIE required a local model computed using a small neighborhood of 25 training samples to achieve comparable performance (with the global model the performance drops from 5.77 to 5.95 degrees on average). In contrast, the IWKR and IWTGP models are more global, since IWTGP takes 800 neighbors into account and IWKR uses all the training data[4].

6.2 HumanEva-I Dataset

HumanEva-I contains synchronized multi-view video and Mocap data. It consists of 3 subjects performing multiple activities: walking, jogging, boxing, throw and catch, and gesturing. We use the histogram of oriented gradient (HoG) features ($\in \mathbb{R}^{270}$) proposed in [5] (we refer to [5] for details[5]). We use training and validations sub-sets of HumanEva-I and only utilize data from 3 color cameras with a total of 9630 image-pose frames for each camera. This is consistent with experiments in [5]. We use half of the data (4815 frames) for training and half (4815 frames) for testing; the test and training data is disjoint. Where fewer, e.g., $n_{\text{tr}} = 500$, training samples are necessary (as in Figure 2) we randomly subsample n_{tr} from the full training set; to alleviate the sampling bias we sample 10 times and average the resulting errors.

The bias in pose estimation can come in (at least) two forms: (1) the training data may simply be biased and, for example, not contain the subject present in

[4] While all the data is used it is dynamically re-weighed based on the importance weight so not all of it is *active* at all times.

[5] We thank the authors for making their features publicly available.

Table 2. Performance on the entire HumanEva-I dataset averaged over all motions.

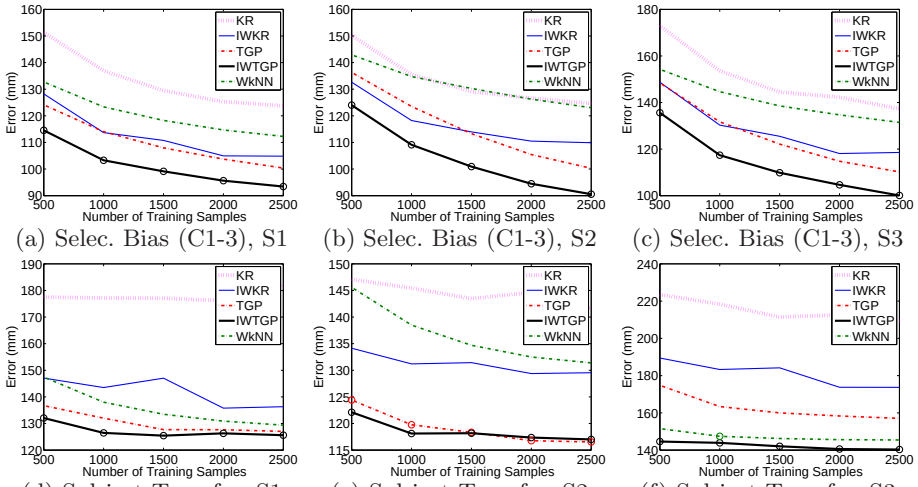
Transfer Type	Subject Train	Test	**IWTGP**	TGP	**IWKR**	KR	WkNN
Selection Bias (C1)	S1,S2,S3	S1	**54.2**	55.1	**71.1**	80.1	70.2
	S1,S2,S3	S2	**52.5**	53.2	**67.6**	75.5	71.5
	S1,S2,S3	S3	**57.5**	57.9	**75.1**	86.0	72.5
Selection Bias (C1-3)	S1,S2,S3	S1	**81.9**	83.9	**101.9**	119.9	94.3
	S1,S2,S3	S2	**72.7**	75.1	**102.7**	120.0	100.7
	S1,S2,S3	S3	**77.2**	86.1	**111.7**	134.8	110.4
Subject Transfer (C1)	S2,S3	S1	**126.2**	126.9	**137.3**	168.4	128.2
	S1,S3	S2	116.7	**116.6**	**130.5**	141.5	130.6
	S1,S2	S3	**140.0**	159.7	**168.4**	209.1	145.5

the test set (we call this *subject transfer*), or (2) the training data may contain data from variety of subjects, motions and cameras, where as at test time only a sub-set of that data is presented at any given time (we call this *selection bias*). To evaluate our methods under such scenarios we propose 3 experiments of interest:

Selection bias (C1): Only camera 1 data is used for training and testing.
Selection bias (C1-3): All camera data is used for training and testing ($3 \times 4815 = 14445$ frames of training and 14445 frames of test data).
Subject transfer (C1): Test subject is not included in training phase.

Error metric: In HumanEva-I pose is encoded by (20) 3D joint markers defined relative to the 'torsoDistal' joint in camera-centric coordinate frame, so $\boldsymbol{y} = [\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(20)}]^\top \in \mathbb{R}^{60}$ and $\boldsymbol{y}^{(i)} \in \mathbb{R}^3$. Error (in mm) for each pose is measured as average Euclidean distance: $Error_{pose}(\widehat{\boldsymbol{y}}, \boldsymbol{y}^*) = \frac{1}{20}\sum_{m=1}^{20} \|\widehat{\boldsymbol{y}}^{(m)} - \boldsymbol{y}^{*(m)}\|$, where $\widehat{\boldsymbol{y}}$ is an estimated pose vector, and $\boldsymbol{y}^*$ is a true pose vector.

Performance: Figure 2 shows the average mean pose estimation error as a function of training set size (averaged over all motions and 10 runs). The graphs clearly show that IWTGP and IWKR outperform their un-weighted counterparts. Moreover, IWTGP overall compares favorably with existing methods in terms of the overall performance. Table 2 shows performance using the entire training set. IWTGP tends to have smaller error compared to all other methods. Note that both the weighted and their un-weighted counterparts use the same parameters and inference procedures; the key difference is in the interest weighting that alters the learning. Moreover, paired t-tests were conducted for all experiments, we observe that about 80% cases the importance weighted methods, IWTGP and IWKR, statistically outperform their non-weighted counterparts at p=0.05 (5%) significance. In certain settings, we see more drastic improvements, e.g., 14% reduction in error in subject transfer with S3 using IWTGP (and 19% using IWKR), or over 10% reduction in error in selection bias (C1-3) with S3. We also see significant improvements on certain specific motions (see supplementary material), where, for example, on gesture motion under selection bias we observe improvement by 22.6 mm (reducing error by 20%) or under subject transfer by 64.5 mm (reducing error by 33%).



(a) Selec. Bias (C1-3), S1 (b) Selec. Bias (C1-3), S2 (c) Selec. Bias (C1-3), S3

(d) Subject Transfer, S1 (e) Subject Transfer, S2 (f) Subject Transfer, S3

Fig. 2. Performance on HumanEva-I dataset illustrated as a function of the number of training samples; we averaged the error over all motions for each subject. Comparable methods according to the paired *t-test* at the significance level 5% are specified by '∘'.

Conclusions: We propose a simple, yet effective, unsupervised method for addressing training set bias through covariate shift adaptation in (structured) prediction problems. As part of our formulation, we also introduce importance weighted variants of kernel regression (IWKR) and twin Gaussian processes (IWTGP) which produce state-of-the-art 3D pose estimation performance on standard datasets (HumanEva-I and Poser [2]). We view our approach as the first step towards eliminating bias in structured prediction problems in vision.

References

1. Agarwal, A., Triggs, B.: Monocular human motion capture with a mixture of regressors. In: CVPR Workshop. (2005)
2. Agarwal, A., B.Triggs: Recovering 3D human pose from monocular images. IEEE Trans. on PAMI **28** (2006) 44—58
3. Bissacco, A., Yang, M., Soatto, S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In: CVPR. (2007) 1–8
4. Bo, L., Sminchisescu, C., Kanaujia, A., Metaxas, D.: Fast algorithms for large scale conditional 3d prediction. In: CVPR. (2008)
5. Bo, L., Sminchisescu, C.: Twin gaussian processes for structured prediction. Int. J. Comput. Vision **87** (2010) 28–52
6. Ek, C., Torr, P., Lawrence, N.: Gaussian process latent variable models for human pose estimation. In: Workshp on ML for Mult. Inter. Volume 4892., LNCS (2007)
7. Ionescu, C., Bo, L., Sminchisescu, C.: Structural svm for visual localization and continuous state estimation. In: ICCV. (2009)
8. Kanaujia, A., Sminchisescu, C., Metaxas, D.: Semi-supervised hierarchical models for 3d human pose reconstruction. In: CVPR. (2007)
9. Navaratnam, R., Fitzgibbon, A., Cipolla, R.: The joint manifold model for semi-supervised multi-valued regression. In: ICCV. (2007)

10. Rosales, R., S.Sclaroff: Learning body pose via specialized maps. In: NIPS. (2002)
11. Salzmann, M., Ek, C.H., Urtasun, R., Darrell, T.: Factorized orthogonal latent spaces. In: AISTATS. (2010)
12. Sigal, L., Memisevic, R., Fleet, D.: Shared kernel information embedding for discriminative inference. In: CVPR. (2009)
13. Shakhnarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: ICCV. Volume 2. (2003) 750—757
14. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Discriminative density propagation for 3d human motion estimation. In: CVPR. (2005)
15. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Learning joint top-down and bottom-up processes for 3d visual inference. In: CVPR. (2006)
16. Urtasun, R., Darrell, T.: Sparse probabilistic regression for activity-independent human pose inference. In: CVPR. (2008)
17. Zhao, X., Ning, H., Liu, Y., Huang, T.S.: Discriminative estimation of 3d human pose using gaussian processes. In: CVPR. (2008)
18. Aytar, Y., Zisserman, A.: Tabula rasa: Model transfer for object category detection. In: ICCV. (2011)
19. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR. (2011)
20. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010)
21. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV. (2009)
22. Torralba, A., Efros, A.: Ubiased look at dataset bias. In: CVPR. (2011)
23. Sigal, L., Black, M.J.: Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In: TR CS-06-08, Brown Univ. (2006)
24. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference **90** (2000)
25. Sigal, L., Balan, A., Black, M.: Combined discriminative and generative articulated pose and non-rigid shape estimation. (2007)
26. de Campos, T., Murray, D.: Regression-based hand pose estimation from multiple cameras. In: CVPR. Volume 1. (2006) 782–789
27. Rosales, R., Athitsos, V., Sigal, L., Scarloff, S.: 3d hand pose reconstruction using specialized mappings. In: ICCV. Volume 1. (2001) 378–385
28. Fanelli, G., Gall, J., Gool, L.V.: Real time head pose estimation with random regression forests. In: CVPR. (2011)
29. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., B. Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS. (2007)
30. Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: NIPS. (2008)
31. Kanamori, T., Hido, S., Sugiyama, M.: A least-squares approach to direct importance estimation. JMLR **10** (2009) 1391–1445
32. Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., Sugiyama, M.: Relative density-ratio estimation for robust distribution comparison. In: NIPS. (2011)
33. Cortes, C., Mansour, Y., Mohri, M.: Learning bounds for importance weighting. In: NIPS. (2010)
34. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
35. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. JMLR **8** (2007) 985–1005