Predictor Combination at Test Time

Kwang In Kim
University of Bath

James Tompkin
Brown University

Christian Richardt
University of Bath

Abstract

We present an algorithm for test-time combination of a set of reference predictors with unknown parametric forms. Existing multi-task and transfer learning algorithms focus on training-time transfer and combination, where the parametric forms of predictors are known and shared. However, when the parametric form of a predictor is unknown, e.g., for a human predictor or a predictor in a precompiled library, existing algorithms are not applicable. Instead, we empirically evaluate predictors on sampled data points to measure distances between different predictors. This embeds the set of reference predictors into a Riemannian manifold, upon which we perform manifold denoising to obtain the refined predictor. This allows our approach to make no assumptions about the underlying predictor forms. Our test-time combination algorithm equals or outperforms existing multi-task and transfer learning algorithms on challenging real-world datasets, without introducing specific model assumptions.

1. Introduction

When a single predictor is insufficient for a task, we can *refine* it based on a set of *reference* predictors with the same input domain. Recent advances in multi-task and transfer learning have shown this by exploiting dependencies across multiple related tasks. These approaches estimate and selectively strengthen pairwise similarities between different predictors. Most work on these problems focuses on the predictor *training phase*, where known parametric representations allow similarities to be measured and enforced (w.r.t. a Euclidean metric). However, these assumptions make it impossible to exploit dependencies between multiple predictors with different representations, such as support vector machines defined on different input feature spaces, or predictors based on precomplied libraries, or even predictors without any explicit functional form, such as human predictors.

We call this problem **predictor combination**: refining a given predictor using a set of reference predictors with unknown form. In this scenario, there is no guarantee that all reference predictors are even relevant to the given task.

We present an algorithm to exploit only the relevant predictors by automatically estimating their dependencies at *test time*. Unlike prior techniques which assume known (and possibly shared) predictor parametric forms, we assume no known parametric form or even a shared parameter space. We posit that reference predictors lie on an underlying manifold M, and that our initial predictor f is a noisy observation of an underlying predictor t on M. We model points (or predictors) on this manifold as non-parametric Gaussian processes (GPs). Then, the similarity between two predictors is obtained as the KL-divergence between their corresponding GPs. This renders M as a Riemannian manifold with the Fisher information metric. Refining the predictor of interest—combining it with our reference predictors—is then manifold denoising: we refine the original noisy predictor through a diffusion process on the reference predictor manifold.

The manifold assumption has been successfully applied to multi-task learning [1, 11, 21]. However, the crucial difference is that we do not use any explicit parametrization, which facilitates combining multiple heterogeneous predictors. Since the distances between different predictors are measured in KL-divergences, it is inherently adaptive to the data generating distributions as similarities between predictors are stressed more in high-density regions. In contrast, in classical parametric models, once the parameters are constructed, the distances between them are agnostic to data distributions. Further, if prediction confidences are available, our GP model provides a natural way to exploit them.

As our problem is to combine multiple existing predictors of unknown form at test time, our approach is categorically different to existing multi-task and transfer learning algorithms. To enable comparison, we conduct experiments in which the parametric forms are explicitly provided to these existing algorithms, but not to our approach. We compare on challenging datasets including human body shape and pose estimation, as well as on multi-task regression benchmark datasets. In this setting, the performance of our approach is comparable to or outperforms existing techniques, even though we make no assumptions about the predictors' parametric forms—even that a parametric form exists at all.

2. Related work

Our algorithm aims to *refine* a predictor f based on a set $\{h^i\}$ of fixed reference predictors at test time; no knowledge of f or $\{h^i\}$ is assumed. This could be regarded as an instance of multi-task learning (MTL), in which all tasks are learned simultaneously (assuming $f = h^1 \in \{h^i\}$). However, our problem differs in that we focus on refining one predictor given the reference predictors. It could also be regarded

as an instance of transfer learning (TL), in which a single task is learned given a fixed set of predictors. However, in our problem, we use multiple reference predictors to refine a given predictor instead of transferring a single reference predictor to a given task domain. With additional assumptions, existing MTL and TL algorithms can be applied to our setting. However, to our knowledge, no previous MTL or TL algorithm has targeted the combination of predictors with unknown form.

That said, our algorithm is motivated by the success of MTL and TL algorithms that directly enforce the similarity between predictor parameters [4, 10, 11, 17, 22, 32, 36]. These methods assume that all predictors (either fixed or not) share a common parametric form, and regularize the estimate by minimizing the pairwise distances between the parameter vectors of the predictors. Assuming that each predictor is linear, i.e., $h^i(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}^i$, the solutions of these algorithms can be obtained by minimizing the joint energy functional

$$\mathcal{E}_{\text{MTL}}(\mathbf{W}) = \sum_{i=1}^{n} \hat{R}(h^i) + \lambda_1 \sum_{i=1}^{n} \|\mathbf{w}^i\|^2 + \lambda_2 \text{tr}[\mathbf{W}^\top \mathbf{L} \mathbf{W}], \tag{1}$$

where $\mathbf{W} = [\mathbf{w}^1, ..., \mathbf{w}^n]$, $\hat{R}(h^i)$ is the training error of h^i, $\text{tr}[\mathbf{A}]$ is the trace of matrix $\mathbf{A}$, and $\mathbf{L}$ is the graph Laplacian encoding pairwise similarities between tasks.

For TL, all predictors $\{h^i\}$ but one ($f = h^1$) are fixed a priori. In this case, there is typically exactly one reference predictor [4, 17, 32]. For MTL, Evgeniou and Pontil [10] constructed the graph Laplacian $\mathbf{L}$ based on a fully connected graph, which is extended to a sparse graph Laplacian-based algorithm that exploits the known task relationships [11]. These methods require prior knowledge about the strengths of relationships between tasks. Pentina et al. [23] addressed this problem by formulating MTL as a curriculum learning problem where the tasks are sequentially learned. The task sequence is then identified during the training by minimizing the upper bound on the generalization error. This class of MTL algorithms can be extended to nonlinear predictors when they are given as kernel-based predictors:

$$h^i(\mathbf{x}) = \Phi(\mathbf{x})^\top \mathbf{w}^i, \tag{2}$$

where Φ is a nonlinear map from $\mathcal{X} \subset \mathbb{R}^n$ to a reproducing kernel Hilbert space (RKHS). Our approach can be regarded as a model-free extension of these algorithms. In our experiments, we show that our algorithm is on par with or outperforms these algorithms, even though they make much stronger assumptions about the parametric form of predictors, while our algorithm is agnostic to their parametric form.

An alternative MTL method to explicitly estimate the relationships between tasks is to identify a common structure among predictors. Typically, such methods assume that a structure is manifested through a low-dimensional latent space that spans all parameter vectors $\{\mathbf{w}^i\}$. For instance, Ruvolo and Eaton [28] learned a low-dimensional projection matrix $\mathbf{P}$ of the parameter matrix $\mathbf{W}$ by minimizing the Frobenius norm of $\mathbf{P}$. Argyriou et al. [3] proposed using a sparse regularizer on $\mathbf{W}$ (which replaces the second regularizer in Equation 1). This has been extended by Kumar and Daumé III to enforce group sparsity [18], by Lozano and Swirszcz for hierarchical group sparsity [20], and has been generalized to (linear or nonlinear) low-rank embeddings to facilitate heterogeneous domain transfer [9, 33]. Bonilla et al. proposed a non-parametric Gaussian process (GP)-based framework that uses shared input kernels across all tasks [7] while Tuitsias and Lázaro-Gredilla used GPs that explicitly model the latent processes shared by all tasks [31]. These structure-based methods have demonstrated good MTL performance. However, they require simultaneous training: access to the training process of individual predictors. As such, their application to our problem domain of refining a predictor given a fixed reference set is not directly possible. Nevertheless, we provide a baseline MTL comparison with Bonilla et al.'s algorithm [7] in the supplemental document.

Another problem strongly related to MTL and TL is domain adaptation: adapting a model trained on a source data distribution to a different target data distribution. This can be approached using TL algorithms, while other (problem-specific) approaches model the change of data distributions [12, 14]. In particular, recent work on object recognition enables test-time adaptation [19, 27]. However, these algorithms focus on modeling and adapting the change of distributions—which is not the case in our combination problem—and are thus complementary to our contribution.

3. Test-time predictor combination

Suppose that we are given a deterministic function $f \in C^\infty(\mathcal{X})$ on the domain $\mathcal{X}$ as an estimate of a target regression function t. The estimate f might be either explicitly constructed by training on a dataset sampled from an underlying probability distribution $p_{\mathcal{X}\times\mathbb{R}}$, or the functional form might be unknown, e.g., if it is a precompiled software package. In both cases, we assume that f can be evaluated for a given set $U = \{\mathbf{x}_1, ..., \mathbf{x}_u\} \subset \mathcal{X}$ of data points. Our predictor combination task is to *refine* f towards the target regression function t based on auxiliary information available on t. Here, we assume that such auxiliary information is available in the form of reference predictors $H = \{h^i\}$. Similarly to f, each element h^i may not have a specific parametric form. Furthermore, it may or may not be related to the underlying ground-truth t or its estimate f. Our goal is first to identify the relevant reference predictors in H (if any), and then to exploit the identified reference predictors to refine f.

3.1. The predictor manifold M

While each reference predictor $h^i \in H$ can be assumed to be an element of $C^\infty(\mathcal{X})$ (i.e., h^i is a deterministic function), we consider the more general case where h^i has a covariance operator $\mathcal{K}^i$. This *probabilistic predictor* can be modeled as a Gaussian process (GP) with a mean function h^i and the

covariance kernel k^i such that $\mathcal{K}^i[g] := \int k^i(\mathbf{x},\mathbf{y})g(\mathbf{y})d\mathbf{y}$ for $g \in C^\infty(\mathcal{X})$. With a slight abuse of notation, we refer to this GP by its mean function h^i ($h^i \in \mathcal{G}$). For deterministic functions h^i, we use the unit covariance kernel $k^i(\mathbf{x},\mathbf{y}) = \delta(\mathbf{x},\mathbf{y})$. As detailed shortly, the function f to be refined will also be modeled as a GP with mean function f and covariance operator $\mathcal{K}^f$. Thus, the deterministic prediction $f(\mathbf{x})$ for an input $\mathbf{x} \in \mathcal{X}$ is obtained by evaluating the majority vote predictor (corresponding to GP f) on $\mathbf{x}$ [23]. Note that the GP model is actually fairly general, and includes non-parametric deterministic estimation as a special case. As demonstrated later, the GP assumption helps especially when the reference predictions are provided with predictive variances.

We now construct our reference manifold M from a subset of $\mathcal{G}$, which has square-integrable mean functions and bounded, non-degenerate covariance operators. Each GP h^i in this set can be *projected* onto M by globally centering and scaling the mean function h^i based on its covariance $\mathcal{K}^i$ ($h^i \in \mathcal{G} \to [h^i] \in M$):

$$\left\langle [h^i], \mathbf{1} \right\rangle_{\mathcal{X}} = 0 \quad \text{and} \quad \left\langle [h^i], (\mathcal{K}^i)^{-1}[h^i] \right\rangle_{\mathcal{X}} = 1, \tag{3}$$

where $\mathbf{1}(\cdot) = 1$ is a function equal to one, and $\langle r,s \rangle_{\mathcal{X}} := \int r(\mathbf{x})s(\mathbf{x})p_{\mathcal{X}}(\mathbf{x})\, d\mathbf{x}$ with $p_{\mathcal{X}}$ the marginal distribution of $p_{\mathcal{X}\times\mathbb{R}}$. As $\mathcal{K}^i$ is a bounded operator, its inverse $(\mathcal{K}^i)^{-1}$ is well-defined. As will become clear later, this normalization enables us to exploit predictors h^i whose scale deviates from the scale of f. Our model assumption is that the projected predictor $[f]$ is given as a *noisy observation* of an underlying process $[t] \in M$. This leads to a strategy to identify the clean solution $[t]$ by denoising $[f]$ *along* M. Before we present our specific denoising algorithm, we first discuss the details of the manifold structure.

Due to the normalization in Equation 3, each point $[h^i] \in M$ actually corresponds to an equivalence class, where GPs $h^k, h^l \in \mathcal{G}$ correspond to $[h^i]$ if they are square-integrable and deviate only at the set of probability zero:

$$\left\langle h^k - h^l, (\mathcal{K}^i)^{-1}(h^k - h^l) \right\rangle_{\mathcal{X}} = 0. \tag{4}$$

Based on this structure, we can identify M with an embedded submanifold of $\mathcal{G}$ ($\imath : h^i \in \mathcal{G} \to [h^i] \in M$). This renders M into a (semi)-Riemannian manifold based on the f-divergences defined originally on $\mathcal{G}$ [2]. A natural f-divergence between two processes h^i and h^j on $\mathcal{G}$ is the Kullback–Leibler (KL) divergence:

$$\mathrm{KL}(h^i \,|\, h^j) = \int \ln\left(\frac{p^i(g)}{p^j(g)}\right) p^i(g) dg, \tag{5}$$

where p^i is the distribution of h^i.[1]

[1] This induces the *Fisher information metric* as a metric in $\mathcal{G}$:

$$g^{\mathcal{G}}(F,H) = \int \frac{dF}{dp}\frac{dH}{dp} p(g) dg, \tag{6}$$

where $F,H \in \mathcal{T}(\mathcal{G})$ lie in the *tangent bundle* $\mathcal{T}(\mathcal{G})$ of $\mathcal{G}$ [16]. The metric g^M on M as a submanifold of $\mathcal{G}$ is then inherited from $g^{\mathcal{G}}$. This opens the possibility of analyzing M (in particular the convergence of the estimate $[f]$ to $[t]$) based on the well-developed methods of *information geometry* [2]. However, we focus on the algorithmic construction, which can be facilitated by using the f-divergence of $\mathcal{G}$ instead of the metric g^M defined on M.

[2] This could lead to a new multi-task learning algorithm, where the set of tasks are jointly denoised by exploiting task dependencies.

3.2. Manifold denoising algorithm

We adopt the manifold denoising approach of Hein and Maier [13], which enables denoising a noisy sample of an underlying manifold M as represented by a point cloud of the *ambient* Euclidean space. We start with the description of the original manifold denoising algorithm [13], and then discuss its application to denoising the predictor $[f]$. We assume that our reference predictor set H is a clean sample and therefore needs no denoising. If H is noisy, H and f can be jointly denoised.[2] For notational convenience, we will use f and h^i to also denote their projections onto M (instead of $[f]$ and $[h^i]$).

Suppose that a set of data points $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_n\} \subset \mathbb{R}^d$ is given as a (noisy) sample of an m-dimensional manifold M embedded in $\mathbb{R}^d$ ($\imath(M) \subset \mathbb{R}^d$). Then, G is a sample from a probability distribution on $\mathbb{R}^d$ which is supported only on $\imath(M)$. The metric of M is induced from $\mathbb{R}^d$. Now we assume that the noise is i.i.d. isotropic Gaussian $\boldsymbol{\epsilon}$ in $\mathbb{R}^d$:

$$\mathbf{g}_i = \imath(\mathbf{o}_i) + \boldsymbol{\epsilon}, \tag{7}$$

where $\{\mathbf{o}_1, \ldots, \mathbf{o}_n\} \subset M$ are the underlying noise-free data points. Hein and Maier's algorithm denoises $G \subset \mathbb{R}^d$ by inducing a diffusion process on M. First, they build a graph Laplacian $\mathbf{L}$ as a discrete approximation of the diffusion generator, the Laplace–Beltrami operator:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}, \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix storing the row sums of the matrix $\mathbf{W}$, i.e. $\mathbf{D}_{ii} = \sum_{j=1} \mathbf{W}_{ij}$, and

$$\mathbf{W}_{ij} = w(\mathcal{D}(\mathbf{g}_i, \mathbf{g}_j), \sigma_f^2), \text{ with} \tag{9}$$

$$w(x,c) = \exp\left(-\frac{x^2}{c}\right), \tag{10}$$

where $\mathcal{D}$ is the Euclidean distance function in $\mathbb{R}^d$, and σ_f is a hyper-parameter.

Given the graph Laplacian $\mathbf{L}$, the denoising algorithm is presented as a differential equation describing the diffusion process on the finite set G of data points:

$$\frac{\partial \mathbf{G}}{\partial t} = -\delta \mathbf{L}\mathbf{G}, \tag{11}$$

where $\mathbf{G} = [\mathbf{g}_1 \ \ldots \ \mathbf{g}_n]^\top$ and $\delta > 0$ is a diffusion constant (hyper-parameter). This equation is nonlinear, as $\mathbf{L}$ is a function of G, which evolves over time. Discretizing Equation 11 in time, we obtain an implicit Euler scheme:

$$\mathbf{G}^{t+1} - \mathbf{G}^t = -\delta \mathbf{L}\mathbf{G}^{t+1}. \tag{12}$$

The solution of Equation 12 (per timestep t) is obtained as the minimizer of the following regularized regression energy [13]:

$$\mathcal{E}(\mathbf{Z}) = \|\mathbf{Z}-\mathbf{G}^t\|_{\mathrm{F}}^2 + \delta\langle \mathbf{Z}, \mathbf{L}\mathbf{Z}\rangle, \tag{13}$$

where $\|\mathbf{A}\|_{\mathrm{F}}$ is the Frobenius norm of matrix $\mathbf{A}$ and $\langle \mathbf{A},\mathbf{B}\rangle := \mathrm{tr}[\mathbf{A}^\top \mathbf{B}]$. The denoising algorithm iterates minimizing the energy $\mathcal{E}$ in Equation 13 until the termination condition is met (e.g., number of iterations). Given the Gaussian noise model, this process directs G (and so $\mathbf{G}$) towards the submanifold $\imath(M)$, and thus eventually makes G lie on $\imath(M)$. As the number n of data points increases, the graph Laplacian $\mathbf{L}$ converges to the corresponding Laplace–Beltrami operator Δ on M [13]. In this case, as the diffusion proceeds with $t \to \infty$, the noisy sample G converges to $\imath(M)$. Note that $\mathbf{L}$ is constructed entirely based on evaluations of the Euclidean distances. Therefore, Equation 13 enables us to perform manifold denoising given only sampled data points in the ambient space $\mathbb{R}^d$, without having to access M directly.

Denoising on the predictor manifold M. Our algorithm extends this strategy to denoising the initial predictor $f^0 := f$. First, we construct $\mathbf{G}$ by stacking f and the reference predictor set H row-wise. Then, our graph Laplacian $\mathbf{L}$ is constructed from the ambient KL-divergences in $\mathcal{G}$ (replacing the Euclidean distance $\mathcal{D}$ in Equation 9).[3] Given the predictors $\mathbf{G}$ and the Laplacian $\mathbf{L}$, denoising is performed by iteratively solving Equation 13. Since H is assumed to be free from noise, only the first row of $\mathbf{G}$ (i.e., f) is updated. Equation 13 then simplifies to:[4]

$$\mathcal{E}(z) = \|z-f^t\|_2^2 + \delta \sum_{h^i \in H} \overline{w}(h^i)\mathrm{KL}(z\,|\,h^i)^2, \tag{14}$$

$$\overline{w}(h^i) = \frac{w(\mathrm{KL}(f^t\,|\,h^i), \sigma_f^2)}{\sum_{h^j \in H} w(\mathrm{KL}(f^t\,|\,h^j), \sigma_f^2)}. \tag{15}$$

The remainder of this section shows that calculating the KL-divergence between z (and f^t) and h^i boils down to calculating the (normalized) correlations between them. This enables exploiting the information on positively correlated references h^i to refine f. To also exploit negatively correlated reference predictors, we augment H by including $-h^i$ for all $h^i \in H$. During the combination, we use either one of h^i or $-h^i$, whichever has the smaller KL-divergence to z.

KL-divergence given unlabeled data points. The denoising cost functional $\mathcal{E}$ in Equation 14 balances the deviation from the previous estimate f^t with the sum of the KL-divergences $\mathrm{KL}(z\,|\,h^i)$ for $h^i \in H$. To facilitate the *comparison* of a deterministic function z with a GP $h^i \in H$, we cast z into a GP by adopting z as its mean function and using the covariance operator $\mathcal{K}^i$. Pentina et al. use this approach to cast a deterministic classification function into a probability distribution [23]. As shown shortly (in Equation 19), this leads to measuring the distance between z and h^i in the function space by weighting the confidence of h^i's prediction. In general, calculating the KL-divergence between two (infinite-dimensional) Gaussian processes is challenging, especially when we do not have direct access to the functional form of f or h^i. However, even in this case, we can still make empirical evaluations of f and h^i on a sample.

We use a set of data points $U = \{\mathbf{x}_1, \ldots, \mathbf{x}_u\}$ sampled from $p_{\mathcal{X}}$ to construct the sample evaluations $\{\mathbf{h}^i := h^i|_U\}$, $\{\mathbf{K}^i := k^i|_{U\times U}\}$, $\mathbf{z} := z|_U$, and $\mathbf{f} := f|_U$. Due to the marginalization property of GPs, once $\mathbf{h}^i$ and its empirical covariance matrix $\mathbf{K}^i$ are given, a consistent infinite-dimensional GP h^i can be *identified* by simply assigning a zero-mean function $\mathbf{h}^i_* := h^i|_{\mathcal{X}\setminus U}$ and unit covariance $\delta(\cdot,\cdot)$ everywhere except for U.[5] With this construction, the KL-divergence between z and h^i can be calculated by decomposing the function variables z and h^i into the observed (on U) and unobserved (on $\mathcal{X}\setminus U$) parts [25, 29]:

$$\begin{aligned}
&\mathrm{KL}(z\,|\,h^i)\\
&= \int\!\!\int p^z(\mathbf{g}) p^z(\mathbf{g}_*|\mathbf{g}) \ln\!\left(\frac{p^z(\mathbf{g}) p^i(\mathbf{g}_*|\mathbf{g})}{p^i(\mathbf{g}) p^i(\mathbf{g}_*|\mathbf{g})}\right) d\mathbf{g}\, d\mathbf{g}_* && (16)\\
&= \int p^z(\mathbf{g}) \ln\!\left(\frac{p^z(\mathbf{g})}{p^i(\mathbf{g})}\right) d\mathbf{g} && (17)\\
&= \frac{1}{2}\left((\mathbf{h}^i-\mathbf{z})^\top (\mathbf{K}^i)^{-1}(\mathbf{h}^i-\mathbf{z}) + \ln\!\left(\frac{\det \mathbf{K}^z}{\det \mathbf{K}^i}\right)\right.\\
&\qquad \left. + \mathrm{tr}[(\mathbf{K}^i)^{-1}\mathbf{K}^z] - u \right) && (18)\\
&= 1 - \mathbf{z}^\top (\mathbf{K}^i)^{-1}\mathbf{h}^i, && (19)
\end{aligned}$$

where p^z is the distribution of z, $\mathbf{g} = g|_U$, and $\mathbf{g}_* = g|_{\mathcal{X}\setminus U}$. In Equations 16 and 17, we used

$$p^z(\mathbf{g}, \mathbf{g}_*) = p^z(\mathbf{g}) p^i(\mathbf{g}_*|\mathbf{g}), \tag{20}$$

taking the covariance operator of z from $\mathcal{K}^i$. Equation 19 is obtained from the normalization condition in Equation 3. Due to this normalization, $\mathbf{z}^\top (\mathbf{K}^i)^{-1}\mathbf{h}^i$ is bounded in $[0,1]$ and $\mathcal{E}$ can be minimized by maximizing the sum of the normalized correlations $\mathbf{z}^\top (\mathbf{K}^i)^{-1}\mathbf{h}^i$ for $h^i \in H$ on the condition that the updated variable z (equivalently, $\mathbf{z}$) does not deviate significantly from the original estimate f^t (the first term in $\mathcal{E}$, Equation 14). The interpretation of the normalization in Equation 19 becomes more straightforward when $\mathbf{K}^i$ is a diagonal matrix containing the predictive variances of h^i on U: the correlation between $\mathbf{z}$ and $\mathbf{h}^i$ is weighted based on confidence in the predictions $\{h^i(\mathbf{x}_j)\}$. This also facilitates the application of the denoising algorithm to

[3] Since the KL-divergence is not symmetric, it is not a proper distance measure. If necessary, the symmetrized KL-divergence can be used, but we simply use the asymmetric Laplacian.

[4] Here we use z instead of f, to stress its role as a variable. Note that the ambient Euclidean metric in Equation 13 is replaced by the KL-divergence.

[5] A less trivial case is when h^i is originally constructed by combining an *explicit* GP prior and a likelihood evaluated on U, rendering it into a predictive distribution defined on the entire domain $\mathcal{X}$.

large-scale datasets U, as the construction and inversion of the dense matrix $\mathbf{K}^i$ can be computationally demanding.

To train $\mathbf{z}$ (and equivalently z), which is originally not in M, we explicitly normalize it before the correlation is calculated (Equation 19, where z replaces f to stress its role as a variable). We minimize the resulting objective function $\mathcal{E}$ using gradient descent with $\mathbf{f}^t = f^t|_U$ initialization:

$$\mathcal{E}(\mathbf{z}) = \|\mathbf{z}-\mathbf{f}^t\|_2^2 + \delta \sum_i \overline{w}(i)(C^i)^2, \text{ with} \tag{21}$$

$$\overline{w}(i) = \frac{w\left(1-(\mathbf{f}^t)^\top(\mathbf{K}^i)^{-1}\mathbf{h}^i, \sigma_f^2\right)}{\sum_j w\left(1-(\mathbf{f}^t)^\top(\mathbf{K}^j)^{-1}\mathbf{h}^j, \sigma_f^2\right)}, \text{ and} \tag{22}$$

$$C^i = \frac{\mathbf{z}^\top \mathbf{S}(\mathbf{K}^i)^{-1}\mathbf{h}^i}{\sqrt{\mathbf{z}^\top \mathbf{S}(\mathbf{K}^i)^{-1}\mathbf{S}\mathbf{z}}}, \tag{23}$$

where $\mathbf{S} = \mathbf{I} - \frac{1}{u}\mathbf{1}\mathbf{1}^\top$, $\mathbf{1} = [1,\ldots,1]^\top$, and $\{\mathbf{h}^i\}$ and $\mathbf{f}^t$ are pre-normalized. Note that calculating the energy $\mathcal{E}$ and its derivative $\partial\mathcal{E}/\partial\mathbf{z}$ both have computational complexity linear in the number u of sample points given the diagonal covariance matrix $\mathbf{K}^i$. In the energy $\mathcal{E}$, the weighting function w measures the similarity between the estimate f^t and each reference h^i based on the sample U. The hyper-parameter δ controls the overall contribution of the regularizer (the second term in Equation 21), while σ_f (Equation 9) determines the relative amount of the contribution of each reference predictor in refining f: when $\sigma_f = \infty$, all references contribute equally to the refinement of f, whereas only the most strongly related references contribute to the regularizer for $\sigma_f \to 0$. In practice, inverting the covariance matrices $\{\mathbf{K}^i\}$ can be ill-conditioned. We explicitly stabilize them by adding $\mathbf{I}$ before the inversion.

Discussion. Our algorithm is unsupervised: it refines the initial predictor f^0 given unlabeled data points U and reference predictors H. However, if the initial estimate f^0 is trained based on labeled data points $S = \{(\mathbf{x}_{u+1}, y_{u+1}), \ldots, (\mathbf{x}_{u+l}, y_{u+l})\}$, the entire process of constructing the noisy sample estimate $\mathbf{f}^0$, and denoising it, can be regarded as semi-supervised learning (see Section 4). We adopt this setting to facilitate fair comparison with other methods as well as for automatic tuning of hyper-parameters.

The effect of using the computationally efficient diagonal covariance $\mathbf{K}^i$ is that our energy functional $\mathcal{E}$ does not take into account the *spatial* smoothness of $\mathbf{z}$. This can be enforced by constructing an additional spatial regularization term based on a domain graph Laplacian $\mathbf{L}_\mathcal{X}$ constructed from the sparsified covariance $\mathbf{K}^i$, as commonly used in semi-supervised learning and spectral clustering [8, 35]:

$$\mathcal{R}(\mathbf{z}) = \delta_\mathrm{L} \mathbf{z}^\top \mathbf{L}_\mathcal{X} \mathbf{z}, \tag{24}$$

with $\mathbf{L}_\mathcal{X} = \mathbf{I} - \mathbf{D}^{-1}\overline{\mathbf{K}}^i$, hyper-parameter δ_L, and $\overline{\mathbf{K}}^i_{jk} = \mathbf{K}^i_{jk}$ if $\|\mathbf{x}_j - \mathbf{x}_k\| \le \theta$ and 0, otherwise, for a threshold parameter θ. Our initial tests indicated that this additional regularizer could help improve performance; however, this requires tuning two more hyper-parameters, δ_L and θ. For simplicity, we thus do not use this approach.

Our algorithm is constructed from a geometric intuition of manifold denoising. In the supplemental material, we present an alternative interpretation based on the assumption that our references are constructed explicitly as GP predictive distributions, i.e., from a PAC-Bayesian perspective.

4. Experiments

As our approach allows combination of predictors of unknown parametric form, existing approaches which require known parametric forms are not applicable. To enable experimental comparison with multi-task or transfer learning, we thus devise a scenario in which existing algorithms are provided with explicit parametric forms while our algorithm is not. To facilitate the objective assessment of our algorithm in this case, we include the training process of f (based on S) in the evaluation of the algorithm. This also facilitates automatic tuning of hyper-parameters σ_f and $\delta_\mathcal{X}$ (Equation 9).[6]

Our setup. Our algorithm starts with an initial target predictor f and the set of reference predictors H, and produces a denoised target predictor f^*. For each problem, the initial estimate f is obtained as a GP regressor with standard Gaussian covariance kernel $k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|/\sigma_\mathcal{X}^2)$ with scale parameter $\sigma_\mathcal{X}$, trained based on a labeled data set $S = \{(\mathbf{x}_{u+1}, y_{u+1}), \ldots, (\mathbf{x}_{u+l}, y_{u+l})\}$. The mean function f is obtained as a minimizer of the energy functional

$$\mathcal{E}_\mathrm{GP}(f) = \left(\sum_{(\mathbf{x},y)\in S} (f(\mathbf{x}) - y)^2\right) + \delta_\mathcal{X} \|f\|_k^2, \tag{25}$$

where $\|f\|_k$ is the reproducing kernel Hilbert space (RKHS) norm of f corresponding to the covariance kernel k [25], and $\delta_\mathcal{X}$ represents the noise model. As our denoising algorithm uses the unlabeled dataset U, the entire training process, including hyper-parameter tuning (using the labeled dataset S), becomes semi-supervised.

Baseline setup. We adapt Evgeniou and Pontil's graph Laplacian-based algorithm [11] ($GL_{\{1,2\}}$) and Pentina et al.'s curriculum learning algorithm (*CL*) [23], plus baseline independent GP predictions (*Ind*). Note that in our predictor combination problem setting, the first approach [11] is equivalent to transfer learning [4, 17, 32], while Pentina et al.'s algorithm corresponds to choosing the best reference in H that minimizes the generalization error bound [23]. We implemented two different versions of Evgeniou and Pontil's algorithm: the first version (GL_1) uses the graph Laplacian (see Equation 1) with the uniform weight matrix $\mathbf{W} = \mathbf{1}\mathbf{1}^\top$, while the second version (GL_2) computes weights using the Euclidean distance between parameter vectors. This can be regarded as a parametric version of our

[6] As our denoising algorithm is unsupervised, in practice the hyper-parameters would be adjusted by the user to suit the application.

KL-divergence-based similarity (see Equations 10 and 14):

$$\mathbf{W}_{ij} = w(\|\mathbf{w}_i - \mathbf{w}_j\|, \sigma_{\mathrm{w}}^2), \tag{26}$$

with σ_{w} being a hyper-parameter.

Datasets. We compute results on four regression datasets: *CAESAR*, *SARCOS*, *MOCAP* and *School*. We report performance for all algorithms with varying numbers of labeled training data points. We repeat each experiment 10 times with different training and test set splits, and average the results. Due to the large number of experiments, we cannot include all results in the main paper. As such, Figure 1 shows two predictor combinations per datasets which produce a large reduction in error. While not all predictor combinations show such marked improvement, our approach in general outperforms or matches state-of-the-art baselines which make additional assumptions. Our supplemental material shows all combinations, including where the combination does not help. Even in these cases, the results demonstrate that combination never degrades the performance compared to the baselines. We also demonstrate the utility of non-parametric predictor combination in the context of facial landmark detection (*Landmarks* dataset) where traditional parametric combination algorithms are futile to apply.

Complexity. Given the initial predictor f, our algorithm iteratively minimizes $\mathcal{E}$ (Equation 21). The complexity of each gradient evaluation is linear in the number u of data points, and the number $|H|$ of reference predictors. For the *SARCOS* dataset, with 44,484 data points and 21 attributes (predictors), each gradient evaluation computation took 10 ms.

4.1. CAESAR dataset

This dataset contains 4,258 3D scans of human bodies along with 6 ground-truth measurements: arm length, age, sitting height, weight, shoulder breadth, and foot length [24, 26]. Each body scan is represented as a 20-dimensional feature vector by fitting a statistical body model [24]. Our goal is to refine the initial *target predictor* f of each body measurement by using the remaining 5 measurements as reference predictors H. This constitutes 6 different predictor combination problems.

For our algorithm, each of the 5 observed measurements is used directly as a reference predictor. The corresponding GPs are constructed by using the unit covariance $\delta(\cdot,\cdot)$ (Section 3.1). This corresponds to the simplest and least restrictive application case, where no model assumption on H is imposed.[7] However, this setting is *not applicable* to baselines *GL*$_1$, *GL*$_2$ and *CL* as they require explicit representations of the reference predictors H. Therefore, for them, the reference predictors $H = \{h^i\}$ are explicitly constructed as GP regressors trained on the observed reference variables.

[7] We provide additional experimental results, where we use simple (or less sophisticated) linear regressors (instead of Gaussian process regressors), which demonstrate that our combination works even in this case.

Results. In realistic applications, not all predictors are relevant. Age, for example, is not strongly correlated with body length measurements. However, two predictors (arm and foot length) benefit significantly from the combinations obtained by our algorithm (Figure 1). Predictor combinations for the other variables are on par with baseline algorithm *Ind*. The other baseline algorithms *GL*$_1$, *GL*$_2$, *CL* show no noticeable improvement over *Ind* for any combination.

4.2. SARCOS dataset

This kinematics dataset contains 44,484 points collected from a robot arm. The input consists of 7 joint positions, 7 velocities and 7 accelerations, and the output consists of 7 torques [34]. The experimental setting is the same as *CAESAR*: we aim to refine the predictor of each output attribute given the remaining 6 attributes as references. For our algorithm, the reference predictors are obtained in the same way as *CAESAR*. For *GL*$_1$, *GL*$_2$ and *CL*, GP regressors are constructed. Due to the large size of the *SARCOS* dataset, training the full GP reference models is infeasible, so we adopt Snelson and Ghahramani's sparse GP approximation [30] using 1000 inducing data points.

Results. Four out of seven predictors significantly benefit from our predictor combinations; we show the fourth and seventh predictors in Figure 1. This is in accordance with the measured (inverse) KL-divergences shown in Table 1: target variables 2, 3, 4 and 7 have particularly small KL-divergences (large $1-\mathrm{KL}$) with each other, which indicates their mutual relevance. The other algorithms show no significant improvement compared to *Ind*.

4.3. MOCAP dataset

Human body poses are captured with an optical marker system across 50,000 data points [5]. Each data point describes the 3D location of 62 skeletal joint locations (i.e. $62\times3=186$ output dimensions). We estimate these joint locations from the 3D locations of five *end effectors* (left/right hand, left/right foot, and head), i.e., a $5\times3=15$ dimensional mid-level representation as inputs. We removed redundant variables from the original 186-dimensional space, leaving an 87-dimensional data representation. We randomly sample eight of these as target predictors. The experimental setting follows *CAESAR* and *SARCOS* except that, for our algorithm, we adopt the explicit GP model assumption for the reference predictors H in the same way as *GL*$_1$, *GL*$_2$ and *CL*. All reference predictors are explicitly constructed based on sparse GP approximation with 1000 inducing data points. This facilitates direct (model-based) comparison. In this setting, our algorithm further benefits from the available predictive variances, which improve the estimation of the KL-divergences. Using GP predictive variances reduced the average error rate by 11.34% from the model-free case of using the unit covariance $\delta(\cdot,\cdot)$. However, this reduction is achieved at the expense of making an explicit model assumption, which may restrict the application domain of our

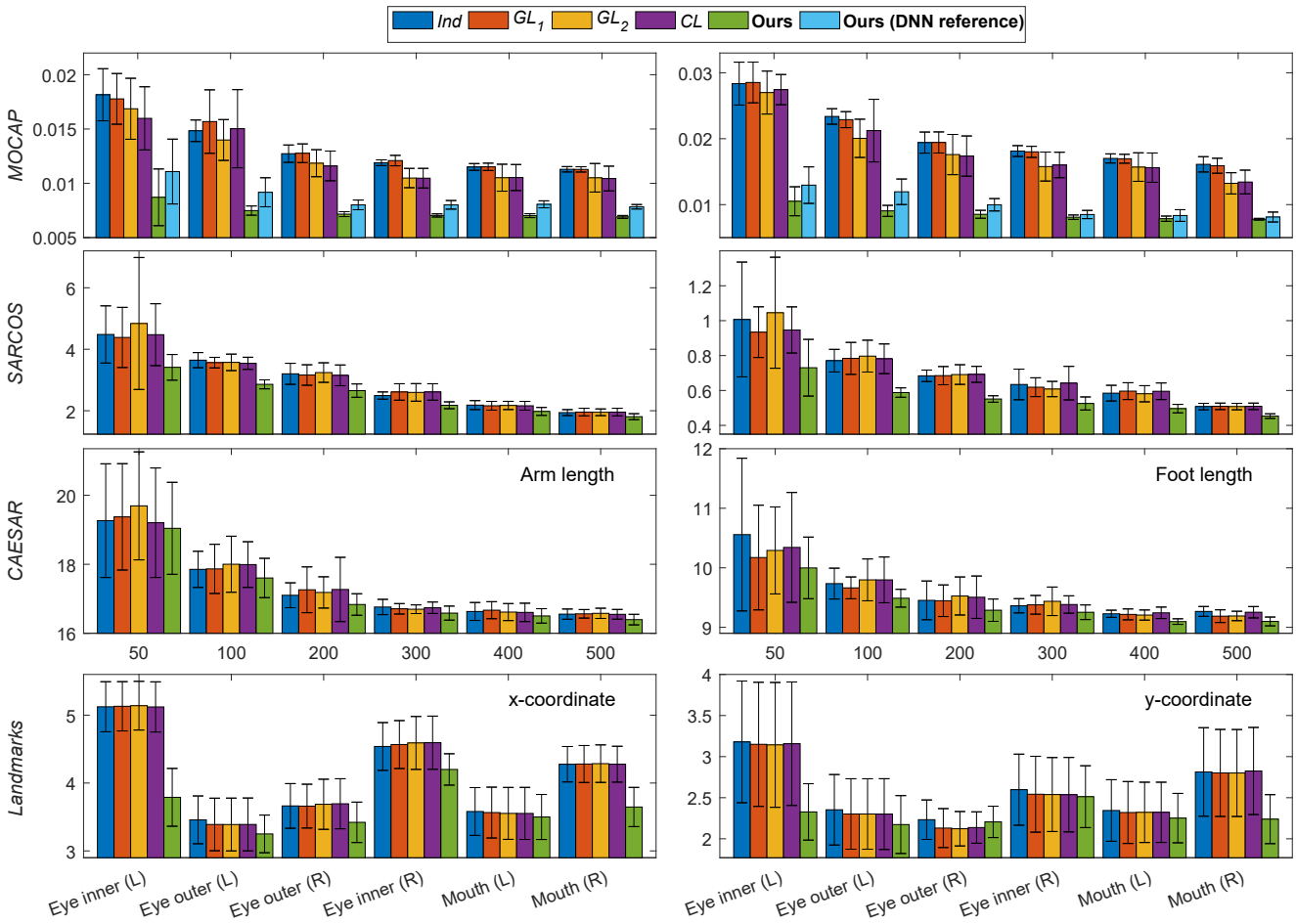

Figure 1. Mean squared error in parameter units (lower is better; error bars are std. deviation). Not all predictor combinations should be expected to be informative, and so for each dataset, we show two predictor combinations with the largest reductions in error over baseline independent predictions. Our supplemental material shows all combinations. In summary, our approach without known parametric form is comparable to or better than existing MTL and TL algorithms with known parametric form. *First three rows*: Regression results across different algorithms on the *MOCAP*, *SARCOS* and *CAESAR* datasets, showing residual error of learning a target predictor f^i given the remaining reference predictors. The horizontal axis shows the number l of labeled data points. *Last row:* Facial landmark detection error, in pixels vs. annotated ground truth. Horizontal axis: indices of six 2D facial landmarks (left: x-coordinates; right: y-coordinates). We compare to: (*Ind*) baseline independent predictions; (GL_1 and GL_2) adaptations of Evgeniou and Pontil [11]; (*CL*) curriculum learning [23].

algorithm (similar to existing algorithms). For comparison, we also provide the combination results obtained based on deep neural network reference predictors optimized based on stochastic gradient descent (Ours–DNN reference).

Results. All eight target predictors show improvement; we show two in Figure 1. GL_1 did not show noticeable improvement over *Ind*, indicating that not all variables are relevant. GL_2 and *CL* show noticeable improvements, but the improvements achieved by our algorithms are much more significant.

4.4. Landmarks dataset

The task is to detect 6 facial landmarks (the corners of both eyes and the mouth) from a face image extracted from the BioID Face Database [15]. Three sliding-window-based non-linear SVM detectors (exploiting facial symmetry) are trained, and the detections are made at the highest responses. We apply our algorithm to detected (x,y)-coordinate values, representing 12 attributes. Detailed description of the SVM detectors, experimental settings, and additional experiments are provided in the supplemental document. Traditional MTL cannot be applied in this setting, as it assumes a shared parametric form for the predictors. We hence apply MTL at the level of the SVM detectors.

Results. For 50 training and 500 test images, over 10 set combinations, we can see that traditional MTL does not help (Figure 1). Enforcing similarity of 'eye-corner' detector and 'mouth' detector actually degrades the performance over individual detectors, as these are not anatomically connected. Our algorithm better exploits predictor dependencies through the detected spatial coordinates.

Table 1. Pairwise 1−KL-divergence values for the *SARCOS* dataset. The target variables 2, 3, 4 and 7 have small KL-divergences leading to mutual improvement by combination.

	1	2	3	4	5	6	7
1	0.00	0.00	0.01	0.26	0.03	0.00	0.17
2	0.00	0.00	0.69	0.33	0.09	0.01	0.41
3	0.01	0.69	0.00	0.47	0.31	0.03	0.54
4	0.26	0.33	0.47	0.00	0.05	0.00	0.93
5	0.03	0.09	0.31	0.05	0.00	0.05	0.09
6	0.00	0.01	0.03	0.00	0.05	0.00	0.01
7	0.17	0.41	0.54	0.93	0.09	0.01	0.00

4.5. School dataset

This dataset consists of examination records of 15,362 students in 139 schools from the Inner London Education Authority [6]. The goal is to predict the exam scores of the students based on 27 input features, such as the year of the exam and gender. Our goal is to estimate the exam scores of each school based on the predictors of the remaining 138 schools as references. This constitutes 139 different combinations of target and reference predictors. We perform experiments on each set trained based on 20 labeled data points, and report the average error rate. Similarly to *MOCAP*, for all combination algorithms, the reference predictors are explicitly constructed as (full) GP predictors.

Results. All four algorithms significantly improved upon *Ind* (Table 2). However, our method shows the least improvement. For this dataset, all tasks are strongly related. All target and reference variables correspond to a single attribute—exam scores—but are sampled from different schools. Thus, only the data sampling distributions are different. This is in contrast to the three other datasets, where each output variable has a different characteristic.

For this dataset, all combination algorithms improve upon independent predictions (*Ind*): Using all parametric references uniformly (GL_1) led to the best results, followed by GL_2, *CL*, and our algorithm. Our algorithm suffered from the lack of data points: the maximum number u of available data points U for each task is 251, with around half of the tasks having less than 100 data points. This demonstrates a limitation of our approach in that data-driven estimation of KL-divergences (Equation 19) can be unreliable versus explicit parametric form modeling (Equation 26). However, even in this case, our result still improves over independent predictions without requiring explicit parametric forms.

5. Discussion

We derived our combination approach from a manifold denoising perspective, which does not model the combination process probabilistically. A probabilistically more rigorous way of combining predictors $H = \{h^i\}$ would look at the joint distribution $p(y_* \,|\, \mathbf{x}_*, f_*) p(f_* \,|\, H)$ (where $f_* := f(\mathbf{x}_*)$): Classical independent Gaussian process (GP)-based regression has a prior distribution $p(f)$ and data likelihood $p(Y \mid X, f)$ corresponding to a training data set $(X,Y) = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$. We predict at $\mathbf{x}_*$ by computing the predictive distribution $p(y_* \mid \mathbf{x}_*, X, Y)$ by marginalizing f_* and f from the joint distribution $p(y_* \,|\, \mathbf{x}_*, f_*) p(f_* \,|\, f) p(f \,|\, X, Y)$. Extending this framework to test-time combination can be achieved by using the distribution $p(y_* \,|\, \mathbf{x}_*, f_*) p(f_* \,|\, f, H) p(f \,|\, X, Y)$ and marginalizing over f_* and f, where $p(f_* \,|\, f, H)$ is a conditional Gaussian distribution. Thus, our approach can be regarded as an indirect way of estimating and using $p(f_* \,|\, f, H)$.

Table 2. Mean squared error (standard deviation in parentheses) on the *School* dataset. All combination approaches improve on independent prediction (*Ind*), although our approach fails to outperform the baselines as the number of unlabeled data points (maximum 251) is too small to reliably estimate KL-divergences.

Ind	GL_1	GL_2	*CL*	Ours
11.86 (2.03)	10.80 (1.82)	11.07 (1.95)	11.18 (1.86)	11.24 (1.86)

6. Conclusions

We presented an algorithm for test-time combination of a set of reference predictors with unknown parametric forms. As there is no guarantee that all reference predictors are relevant to a given task, our algorithm exploits only the relevant predictors by automatically estimating their dependencies at test time. Then, the target predictor is refined using manifold denoising. This makes our algorithm independent of the parametric form of the underlying predictors, unlike existing multi-task and transfer learning algorithms. Crucially, this uniquely enables our algorithm to refine predictors that lack any parametric form, such as human predictors.

Existing algorithms cannot be applied to test-time combination when the parametric forms of predictors are not known. For comparison, when we prepare experiments in which we provide the parametric form to existing multi-task and transfer learning algorithms, our approach is competitive or superior even when it does not know the parametric form. For this reason, our algorithm is more flexible, more versatile, and has wider application potential than existing multi-task and transfer learning algorithms.

Acknowledgments. We thank the anonymous reviewer who suggested the probabilistic interpretation of our approach. This work was funded by EPSRC grants EP/M00533X/2 and EP/M023281/1.

References

[1] A. Agarwal, S. Gerber, and H. Daume. Learning multiple tasks using manifold regularization. In *NIPS*, pages 46–54, 2010.

[2] S. Amari and H. Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2007.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3), 2008.

[4] Y. Aytar and A. Zisserman. Tabula rasa: model transfer for object category detection. In *ICCV*, 2011.

[5] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, pages 1092–1099, 2011.

[6] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *JMLR*, 4, 2003.

[7] E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task Gaussian process prediction. In *NIPS*, pages 153–160, 2008.

[8] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2010.

[9] W.-Y. Chen, T.-M. H. Hsu, Y.-H. H. Tsai, Y.-C. F. Wang, and M.-S. Chen. Transfer neural trees for heterogeneous domain adaptation. In *ECCV*, 2016.

[10] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *KDD*, pages 109–117, 2004.

[11] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *JMLR*, 6:615–637, 2005.

[12] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[13] M. Hein and M. Maier. Manifold denoising. In *NIPS*, pages 561–568, 2007.

[14] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pages 601–608, 2006.

[15] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the Hausdorff distance. In *Audio- and Video-Based Biometric Person Authentication*, pages 90–95, 2001.

[16] J. Jost. *Riemannian Geometry and Geometric Analysis*. Springer, New York, 2011.

[17] A. Kovashka and K. Grauman. Attribute adaptation for personalized image search. In *ICCV*, pages 3432–3439, 2013.

[18] A. Kumar and H. Daumé III. Learning task grouping and overlap in multi-task learning. In *ICML*, pages 1383–1390, 2012.

[19] E. Levinkov and M. Fritz. Sequential Bayesian model update under structured scene prior for semantic road scenes labeling. In *ICCV*, pages 1321–1328, 2013.

[20] A. Lozano and G. Swirszcz. Multi-level lasso for sparse multi-task regression. In *ICML*, 2012.

[21] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE TIP*, 22(2):523–536, 2013.

[22] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[23] A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *CVPR*, pages 5492–5500, 2015.

[24] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3D human modeling. *Pattern Recognition*, 2017.

[25] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[26] K. Robinette, H. Daanen, and E. Paquet. The CAESAR project: a 3-D surface anthropometry survey. In *3-D Digital Imaging and Modeling*, 1999.

[27] A. Royer and C. H. Lampert. Classifier adaptation at prediction time. In *CVPR*, pages 1401–1409, 2015.

[28] P. Ruvolo and E. Eaton. ELLA: An efficient lifelong learning algorithm. *JMLR W&CP (Proc. ICML)*, 28, 2013.

[29] M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *JMLR*, 3, 2002.

[30] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *NIPS*, 2006.

[31] M. K. Titsias and M. Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *NIPS*, pages 2339–2347, 2011.

[32] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: learning categories from few examples with multi model knowledge transfer. In *CVPR*, pages 3081–3088, 2010.

[33] Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*, pages 5081–5090, 2016.

[34] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space. In *ICML*, pages 1079–1086, 2000.

[35] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[36] F. Wang, X. Wang, and T. Li. Semi-supervised multi-task learning with task regularizations. In *ICDM*, pages 562–568, 2009.