Combining Task Predictors via Enhancing Joint Predictability

Kwang In Kim[1][0000−0002−6470−4571], Christian Richardt[2][0000−0001−6716−9845], and Hyung Jin Chang[3][0000−0001−7495−9677]

[1] UNIST, Korea [2] University of Bath, UK [3] University of Birmingham, UK

Abstract. Predictor combination aims to improve a (target) predictor of a learning task based on the (reference) predictors of potentially relevant tasks, without having access to the internals of individual predictors. We present a new predictor combination algorithm that improves the target by i) measuring the relevance of references based on their capabilities in predicting the target, and ii) strengthening such estimated relevance. Unlike existing predictor combination approaches that only exploit pairwise relationships between the target and each reference, and thereby ignore potentially useful dependence among references, our algorithm *jointly* assesses the relevance of all references by adopting a Bayesian framework. This also offers a rigorous way to automatically select only relevant references. Based on experiments on seven real-world datasets from visual attribute ranking and multi-class classification scenarios, we demonstrate that our algorithm offers a significant performance gain and broadens the application range of existing predictor combination approaches.

1 Introduction

Many practical visual understanding problems involve learning multiple tasks. When a *target predictor*, e.g. a classification or a ranking function tailored for the task at hand, is not accurate enough, one could benefit from knowledge accumulated in the predictors of other tasks (*references*). The ***predictor combination* problem** studied by Kim et al. [11] aims to improve the target predictor by exploiting the references without requiring access to the internals of any predictors or assuming that all predictors belong to the same class of functions. This is relevant when the forms of predictors are not known (e.g. precompiled binaries) or the best predictor forms differ across tasks. For example, Gaussian process rankers [9] trained on ResNet101 features [7] are identified as the best for the main task, e.g. for image frame retrieval, while convolutional neural networks are presented as a reference, e.g. classification of objects in images. In this case, existing transfer learning or multi-task learning approaches, such as a parameter or weight sharing, cannot be applied directly.

Kim et al. [11] approached this predictor combination problem for the first time by nonparametrically accessing all predictors based on their evaluations on given datasets, regarding each predictor as a Gaussian process (GP) estimator. Assuming that the target predictor is a noisy observation of an underlying

ground-truth predictor, their algorithm projects all predictors onto a Riemannian manifold of GPs and denoises the target by simulating a diffusion process therein. This approach has demonstrated a noticeable performance gain while meeting the challenging requirements of the predictor combination problem. However, it leaves three possibilities to improve. Firstly, this algorithm is inherently (pairwise) metric-based and, therefore, it can model and exploit only pairwise relevance of the target and each reference, while relevant information can lie in the relationship between multiple references. Secondly, this algorithm assumes that all references are noise-free, while in practical applications, the references may also be trained based on limited data points or weak features and thus they can be imperfect. Thirdly, as this algorithm uses the metric defined between GPs, it can only combine one-dimensional target and references.

In this paper, we propose a new predictor combination algorithm that overcomes these three challenges. The proposed algorithm builds on the manifold denoising framework [11] but instead of their metric diffusion process, we newly pose the predictor denoising as an averaging process, which *jointly* exploits *full dependence* of the references. Our algorithm casts the denoising problem into 1) measuring the *joint* capabilities of the references in predicting the target, and 2) optimizing the target as a variable to enhance such prediction capabilities. By adopting Bayesian inference under this setting, identifying relevant references is now addressed by a rigorous Bayesian relevance determination approach. Further, by denoising *all* predictors in a single unified framework, our algorithm becomes applicable even for imperfect references. Lastly, our algorithm can combine multi-dimensional target and reference predictors, e.g. it can improve multi-class classifiers based on one-dimensional rank predictors. Experiments on *relative attribute* ranking and multi-class classification demonstrate that these contributions individually and collectively improve the performance of predictor combination and further extend the application domain of existing predictor combination algorithms.

Related work. Transfer learning (TL) aims to solve a given learning problem by adapting a source model trained on a different problem [16]. Predictor combination can be regarded as a specific instance of TL. However, unlike predictor combination algorithms, traditional TL approaches improve or newly train predictors of *known* form. Also, most existing algorithms assume that the given source is relevant to the target and, therefore, they do not explicitly consider identifying relevant predictors among many (potentially irrelevant) source predictors.

Another related problem is multi-task learning (MTL), which learns predictors on multiple problems at once [1,2]. State-of-the-art MTL algorithms offer the capability of automatically identifying relevant task groups when not all tasks and the corresponding predictors are mutually relevant. For example, Argyriou et al. [1] and Gong et al. [6], respectively, enforced the sparsity and low-rank constraints in the parameters of predictors to make them aggregate in relevant task groups. Passos et al. [18] performed explicit task clustering ensuring that all tasks (within a cluster) that are fed to the MTL algorithm are relevant. More recently, Zamir et al. [26] proposed to discover a hypergraph that reveals the interdependence of multiple tasks and facilitates transfer of knowledge across relevant tasks.

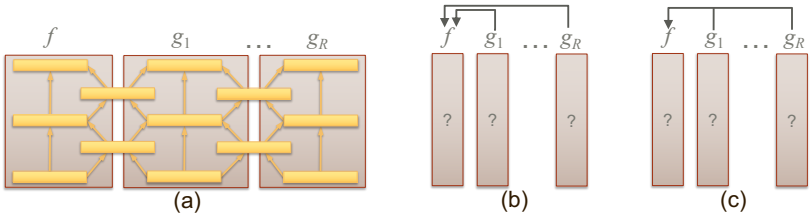

Fig. 1. Illustration of predictor combination algorithms: (a) MTL simultaneously exploits all references $\{g_1,...,g_R\}$ to improve the target predictor f, e.g. by sharing neural network layers. However, they require access to the internals of predictors [2]. (b) Kim et al.'s predictor combination is agnostic to the forms of individual predictors [11] but exploits only pairwise relationships. (c) Our algorithm combines the benefits of both, jointly exploiting all references without requiring their known forms.

While our approach has been motivated by the success of TL and MTL approaches, these approaches are not directly applicable to predictor combination as they share knowledge across tasks via the internal parametric representations [1,6,18] and/or shared deep neural network layers of all predictors (e.g. via shared encoder readouts [26]; see Fig. 1). A closely related approach in this context is Mejjati et al.'s nonparametric MTL approach [13]. Similar to Kim et al. [11], this algorithm assesses predictors based on their sample evaluations, and it (nonparametrically) measures and enforces pairwise statistical dependence among predictors. As this approach is agnostic to the forms of individual predictors, it can be adapted for predictor combination. However, this algorithm shares the same limitations: it can only model pairwise relationships. We demonstrate experimentally that by modeling the joint relevance of all references, our algorithm can significantly outperform both Kim et al.'s original predictor combination algorithm [11] adapted to ranking [10], and Mejjati et al.'s MTL algorithm [13].

2 The predictor combination problem

Suppose we have an *initial predictor* $f^0\colon \mathcal{X}\to\mathcal{Y}$ (e.g. a classification, regression, or ranking function) of a task. The goal of predictor combination is to improve the *target predictor* f^0 based on a set of *reference predictors* $\mathcal{G}=\{g_i\colon \mathcal{X}\to\mathcal{Y}_i\}_{i=1}^R$. The internal structures of the target and reference predictors are unknown and they might have different forms. Crucial to the success of addressing this seriously ill-posed problem is to determine which references (if any) within $\mathcal{G}$ are *relevant* (i.e. useful in improving f^0), and to design a procedure that fully exploits such relevant references without requiring access to the internals of f^0 and $\mathcal{G}$.

Kim et al.'s original predictor combination (*OPC*) [11] approaches this problem by 1) considering the initial predictor f^0 as a noisy estimate of the underlying ground-truth f_{GT}, and 2) assuming f_{GT} and $\mathcal{G}$ are structured such that they all lie on a low-dimensional predictor manifold $\mathcal{M}$. These assumptions enable predictor combination to be cast as well-established *Manifold Denoising*, where one iteratively denoises points on $\mathcal{M}$ via simulating the diffusion process therein [8].

The model space $\mathcal{M}$ of *OPC* consists of Bayesian estimates: each predictor in $\mathcal{M}$ is a GP predictive distribution of the respective task. The natural metric $g_{\mathcal{M}}$ on $\mathcal{M}$, in this case, is induced from the Kullback-Leibler (KL) divergence D_{KL} between probability distributions. Now further assuming that all reference predictors are noise-free, their diffusion process is formulated as a time-discretized evolution of f^t on $\mathcal{M}$: Given the solution f^t at time t and noise-free references $\mathcal{G}$, the new solution f^{t+1} is obtained by minimizing the energy

$$\mathcal{E}_{\mathrm{O}}(f) = D^2_{\mathrm{KL}}(f \,|\, f^t) + \lambda_{\mathrm{O}} \sum_{i=1}^{R} w_i D^2_{\mathrm{KL}}(f \,|\, g_i), \tag{1}$$

where $w_i = \exp(-D^2_{\mathrm{KL}}(f^t \,|\, g_i)/\sigma^2_{\mathrm{O}})$ is inversely proportional to $D_{\mathrm{KL}}(f^t \,|\, g_i)$, and λ_{O} and σ^2_{O} are hyperparameters. Our supplemental document presents how the iterative minimization of $\mathcal{E}_{\mathrm{O}}$ is obtained by discretizing the diffusion process on $\mathcal{M}$.

In practice, it is infeasible to directly optimize functions, which are infinite-dimensional objects. Instead, *OPC* approximates all predictors $\{f, \mathcal{G}\}$ via their evaluations on a test dataset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and optimizes the sample f-evaluation $\mathbf{f} = f|_X := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ based on the sample references $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_R\}$ with $\mathbf{g}_i = g_i|_X$.

At each time step, the relevance of a reference is automatically determined based on its KL-divergence to the current solution: g_i is considered relevant when $D_{\mathrm{KL}}(f^t \,|\, g_i)$ is small. Then, throughout the iteration, *OPC* robustly denoises f by gradually putting more emphasis on highly relevant references while ignoring outliers. This constitutes the first predictor combination algorithm that improves the target predictor without requiring any known forms of predictors (as the KL-divergences are calculated purely based on predictor evaluations). However, Eq. 1 also highlights the limitations of this approach: it exploits only pairwise relationships between the target predictor and individual references, ignoring the potentially useful information that lies in the dependence between references.

Toy problem 1. Consider two references, $\{\mathbf{g}_1, \mathbf{g}_2\} \subset \mathbb{R}^{100}$, constructed by uniformly randomly sampling from $\{0, 1\}$. Here, $\{\mathbf{g}_1, \mathbf{g}_2\}$ are regarded as the means of GP predictive distributions with unit variances. We define the ground-truth target as their difference: $\mathbf{f}_{\mathrm{GT}} = \mathbf{g}_1 - \mathbf{g}_2$. By construction, $\mathbf{f}_{\mathrm{GT}}$ is determined by the *relationship* between the references. Now we construct the initial noisy predictor $\mathbf{f}^0$ by adding independent Gaussian noise with standard deviation 1 to $\mathbf{f}_{\mathrm{GT}}$, achieving the rank accuracy of 0.67 (see Sec. 4 for the definition of the visual attribute ranking problem). In this case, applying *OPC* minimizes $\mathcal{E}_{\mathrm{O}}$ (Eq. 1) but shows insignificant performance improvement as no information on $\mathbf{f}_{\mathrm{GT}}$ can be gained by assessing the relevance of the references individually (Table 1). While this problem has been well-studied in existing MTL and TL approaches, the application of these techniques for predictor combination is not straightforward as they require simultaneous training [6,18] and/or shared predictor forms [26]. Another limitation is that *OPC* requires that all predictions are one-dimensional (i.e. $\mathcal{Y}_i \subset \mathbb{R}$). Therefore, it is not capable of, for example, improving the multi-class classification predictor $\mathbf{f}^0$ given the references constructed for ranking tasks.

Table 1. Accuracies of Kim et al.'s original (*OPC*) [11], and our linear (*LPC*) and non-linear (*NPC*) predictor combination algorithms introduced in Section 3, for illustrative toy problems. $\mathbf{g}_1$ and $\mathbf{g}_2$ are random binary vectors while $\mathbf{f}^0$'s are noisy observations of the corresponding ground-truth predictors $\mathbf{f}_{\mathrm{GT}}$'s.

Toy problem	$\mathbf{f}^0$	*OPC* [11] (Eq. 1)	*LPC* (Eq. 7)	*NPC* (Eq. 13)
1: $\mathbf{f}_{\mathrm{GT}}=\mathbf{g}_1-\mathbf{g}_2$	67.14	67.24	**100**	**100**
2: $\mathbf{f}_{\mathrm{GT}}=\mathrm{XOR}(\mathbf{g}_1,\mathbf{g}_2)$	74.08	74.11	74.24	**100**

3 Joint predictor combination algorithm

Our algorithm takes deterministic predictors instead of Bayesian predictors (i.e. GP predictive distributions) as in *OPC*. When Bayesian predictors are provided as inputs, we simply take their means and discard the predictive variances. This design choice offers a wider range of applications as most predictors – including deep neural networks and support vector machines (SVMs) – are presented as deterministic functions, at the expense of not exploiting potentially useful predictive uncertainties. This assumption has also been adopted by Kim and Chang [10]. Under this setting, our model space is a sub-manifold $\mathcal{M}$ of L^2 space where each predictor has zero mean and unit norm:

$$\forall f\in\mathcal{M}.\quad \int f(\mathbf{x})\mathrm{d}P(\mathbf{x})=0\quad\text{and}\quad \langle f,f\rangle=1, \tag{2}$$

where $\langle f,g\rangle := \int f(\mathbf{x})g(\mathbf{x})dP(\mathbf{x})$ and $P(\mathbf{x})$ is the probability distribution of $\mathbf{x}$. This normalization enables scale and shift-invariant assessment of the relevance of references. The Riemannian metric $g_{\mathcal{M}}$ on $\mathcal{M}$ is defined as the *pullback* metric of the ambient L^2 space: when $\mathcal{M}$ is embedded into L^2 via the embedding $\imath$, $g_{\mathcal{M}}(a,b) := \langle \imath(a),\imath(b)\rangle$. *OPC* (Eq. 1) can be adapted for $\mathcal{M}$ by iteratively maximizing the objective $\mathcal{O}_{\mathrm{O}}$ that replaces the KL-divergence D_{KL} with $g_{\mathcal{M}}(\cdot,\cdot)$:

$$\mathcal{O}_{\mathrm{O}}(f)=g_{\mathcal{M}}(f,f^t)^2+\lambda_{\mathrm{O}}\sum_{i=1}^{R}w_i g_{\mathcal{M}}(f,g_i)^2. \tag{3}$$

For simplicity of exposition, we here assume that the output space is one-dimensional (i.e. $\mathcal{Y}_i=\mathbb{R}$). In Sec. 4, we show how this framework can be extended to multi-dimensional outputs such as for multi-class classification.

The averaging process on $\mathcal{M}$. Both *OPC* (Eq. 1) and its adaptation to our model space (Eq. 3) can model only the pairwise relationship between the target f and each reference $g_i\in\mathcal{G}$, while ignoring the dependence present across the references (*joint relevance* of $\mathcal{G}$ on f). We now present a general framework that can capture such joint relevance by iteratively maximizing the objective

$$\mathcal{O}_{\mathrm{J}}(f)=\langle \imath(f),\imath(f^t)\rangle^2+\lambda_{\mathrm{J}}\langle \imath(f),\mathcal{K}[\imath(f)]\rangle, \tag{4}$$

where $\lambda_{\mathrm{J}}\geq 0$ is a hyperparameter. The linear, non-negative definite averaging operator $\mathcal{K}\colon \imath(\mathcal{M})\to\imath(\mathcal{M})$ is responsible to capture the joint relevance of $\mathcal{G}$ on f. Depending on the choice of $\mathcal{K}$, $\mathcal{O}_{\mathrm{J}}$ can accommodate a variety of predictor combination scenarios, including $\mathcal{O}_{\mathrm{O}}$ as a special case for $\mathcal{K}[\imath(f)]=\sum_{i=1}^{R}\imath(g_i)w_i\langle \imath(f),\imath(g_i)\rangle$.

3.1 Linear predictor combination (LPC)

Our linear predictability operator $\mathcal{K}_{\mathrm{L}}$ is defined as[1]

$$\mathcal{K}_{\mathrm{L}}[\imath(f)] = \sum_{i,j=1}^{R} \imath(g_i) C^{-1}_{[i,j]} \langle \imath(g_j), \imath(f) \rangle \tag{5}$$

using the *correlation matrix* $C_{[i,j]} = \langle \imath(g_i), \imath(g_j) \rangle$. Interpreting $\mathcal{K}_{\mathrm{L}}$ becomes straightforward when substituting $\mathcal{K}_{\mathrm{L}}$ into the second term of $\mathcal{O}_{\mathrm{J}}$ (Eq. 4):

$$\langle \imath(f), \mathcal{K}_{\mathrm{L}}[\imath(f)] \rangle = \mathbf{c}^{\top} C^{-1} \mathbf{c}, \tag{6}$$

where $\mathbf{c} = [\langle \imath(f), \imath(g_1) \rangle, \ldots, \langle \imath(f), \imath(g_R) \rangle]^{\top}$. As each predictor in $\mathcal{M}$ is centered and normalized, all diagonal elements of the correlation matrix C are 1. The off-diagonal elements of C then represent the dependence among the references, making $\langle \imath(f), \mathcal{K}_{\mathrm{L}}[\imath(f)] \rangle$ a measure of *joint correlation* between f and $\mathcal{G} = \{g_i\}_{i=1}^{R}$.

In practice, f and $\{g_i\}$ might not be originally presented as embedded elements $\imath(f)$ and $\{\imath(g_i)\}$ of $\mathcal{M}$: i.e. they are not necessarily centered or normalized (Eq. 2). Also, as in the case of *OPC*, it would be infeasible to manipulate infinite-dimensional functions directly. Therefore, we also adopt sample approximations $\{\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_R\}$ and explicitly project them onto $\mathcal{M}$ via normalization: $\mathbf{f} \to \overline{\mathbf{f}} := \frac{C_N \mathbf{f}}{\|C_N \mathbf{f}\|}$, where $C_N = \mathbf{1}_{N \times N}/N$, $\mathbf{1}_{N \times N}$ is an $N \times N$ matrix of ones, for the sample size $N = |X|$. For this scenario, we obtain our linear predictor combination (*LPC*) algorithm by substituting Eq. 5 into Eq. 4, and replacing f, f^t, and g_j by $\overline{\mathbf{f}}$, $\overline{\mathbf{f}}^t$, and $\overline{\mathbf{g}}_j$, respectively:

$$\mathcal{O}_{\mathrm{L}}(\mathbf{f}) = \frac{(\mathbf{f}^{\top} \overline{\mathbf{f}}^t)^2}{\mathbf{f}^{\top} C_N \mathbf{f}} + \lambda_{\mathrm{J}} \mathcal{P}_{\mathrm{L}}, \tag{7}$$

where $\mathcal{P}_{\mathrm{L}} = \frac{\mathbf{f}^{\top} Q \mathbf{f}}{\mathbf{f}^{\top} C_N \mathbf{f}}$, $Q = G(G^{\top}G)^{-1}G$, and $G = [\overline{\mathbf{g}}_1, \ldots, \overline{\mathbf{g}}_R]$. Here, we pre-projected $\mathcal{G}$ and $\mathbf{f}^t$ onto $\mathcal{M}$ while $\mathbf{f}$ is explicitly projected in Eq. 7. Note that our goal is not to simply calculate $\mathcal{P}_{\mathrm{L}}$ for a fixed $\mathbf{f}$, but to optimize $\mathbf{f}$ while enhancing $\mathcal{P}_{\mathrm{L}}$.

Exploiting the joint relevance of references, *LPC* can provide significant accuracy improvements over *OPC*. For example, *LPC* can generate perfect predictions in Toy Problem 1 (Table 1). However, its capability in measuring the joint relevance is limited to linear relationships only. This can be seen by rewriting $\mathcal{P}_{\mathrm{L}}$ explicitly in $\mathbf{f}$ and G:

$$\mathcal{P}_{\mathrm{L}} = \frac{\mathbf{f}^{\top} Q \mathbf{f}}{\mathbf{f}^{\top} C_N \mathbf{f}} = 1 - \frac{\sum_{i=1}^{N} (\mathbf{f}_i - q(G_{[i,:]}))^2}{\sum_{i=1}^{N} (\mathbf{f}_i - \sum_{j=1}^{n} \mathbf{f}_j / N)^2}, \tag{8}$$

where $G_{[i,:]}$ represents the i-th row of G, and $q(\mathbf{a}) = \mathbf{w}_q^{\top} \mathbf{a}$ is the linear function whose weight vector $\mathbf{w}_q = (G^{\top}G)^{-1} G \mathbf{f}$ is obtained from least-squares regression that takes the reference matrix G as training input and the target predictor variable $\mathbf{f}$ as corresponding labels. Then, $\mathcal{P}_{\mathrm{L}}$ represents the normalized prediction accuracy: the normalizer $\mathbf{f}^{\top} C_N \mathbf{f}$ is simply the variance of $\mathbf{f}$ elements. For this

[1] Here, the term 'linear' signifies the capability of $\mathcal{K}_{\mathrm{L}}$ to capture the linear dependence of references, independent of $\mathcal{K}_{\mathrm{L}}$ being a linear operator as well.

reason, we call $\mathcal{P}_{\mathrm{L}}$ the (linear) *predictability* of G (and equivalently of $\mathcal{G}$) on $\mathbf{f}$. It takes the maximum value of 1 when the linear prediction (made based on G) perfectly agrees with $\mathbf{f}$ when normalized, and it attains the minimum value 0 when the prediction is no better than taking the mean value of $\mathbf{f}$, in which case the mean squared error becomes the variance. Figure 1 illustrates our algorithm in comparison with MTL and *OPC*.

Toy problem 2. Under the setting of Toy problem 1, when the target $\mathbf{f}_{\mathrm{GT}}$ is replaced by a variable that is nonlinearly related to the references, e.g. using the logical exclusive OR (XOR) of $\mathbf{g}_1$ and $\mathbf{g}_2$, *LPC* fails to give any noticeable accuracy improvement compared to the baseline $\mathbf{f}^0$.

3.2 Nonlinear predictor combination (NPC)

Our final algorithm measures the relevance of $\mathcal{G}$ on $\mathbf{f}$ by predicting $\mathbf{f}$ via Gaussian process (GP) estimation. We use the standard zero-mean Gaussian prior and an i.i.d. Gaussian likelihood with noise variance σ^2 [19]. The resulting prediction is obtained as a Gaussian distribution with mean $\mathbf{m}_{\mathbf{f}}$ and covariance $C_{\mathbf{f}}$:

$$\mathbf{m}_{\mathbf{f}} = K(K+\sigma^2 I)^{-1}\mathbf{f}, \;\; C_{\mathbf{f}} = K - K(K+\sigma^2 I)^{-1}K, \tag{9}$$

where $K \in \mathbb{R}^{N\times N}$ is defined using the covariance function $k\colon \mathbb{R}^R \times \mathbb{R}^R \to \mathbb{R}$:

$$K_{[i,j]} = k(G_{[i,:]}, G_{[j,:]}) := \exp\left(-\frac{\|G_{[i,:]} - G_{[j,:]}\|^2}{\sigma_k^2}\right). \tag{10}$$

Now we refine the linear predictability $\mathcal{P}_{\mathrm{L}}$ by replacing $q(G_{[i,:]})$ in Eq. 8 with the corresponding predictive mean $[\mathbf{m}_{\mathbf{f}}]_i$ (where $[\mathbf{a}]_i$ is the i-th element of vector $\mathbf{a}$):

$$\mathcal{P}_{\mathrm{N}} = \frac{\mathbf{f}^\top Q' \mathbf{f}}{\mathbf{f}^\top C_N \mathbf{f}} = 1 - \frac{\sum_{i=1}^N ([\mathbf{f}]_i - [\mathbf{m}_{\mathbf{f}}]_i)^2}{\sum_{i=1}^N ([\mathbf{f}]_i - \sum_{j=1}^N [\mathbf{f}]_j/N)^2}, \tag{11}$$

where Q' is a positive definite matrix that replaces Q in Eq. 8:

$$Q' = C_N\big(2K(K+\sigma^2 I)^{-1} - (K+\sigma^2 I)^{-1}KK(K+\sigma^2 I)^{-1}\big)C_N. \tag{12}$$

The matrix Q' becomes Q when the kernel $k(\mathbf{a},\mathbf{b})$ is replaced by the standard dot product $k'(\mathbf{a},\mathbf{b}) = \mathbf{a}^\top \mathbf{b}$. Note that the noise level σ^2 should be strictly positive; otherwise, $\mathbf{f}_i = [\mathbf{m}_{\mathbf{f}}]_i$ for all $i \in \{1,\ldots,N\}$, and therefore $\mathcal{P}_{\mathrm{N}} = 1$ for any $\mathbf{f}$. This means the resulting GP model perfectly overfits to $\mathbf{f}$ and all references are considered perfectly relevant regardless of the actual values of G and $\mathbf{f}$.

Computational model. Explicitly normalizing $\mathbf{f}$ ($\mathbf{f} \to \bar{\mathbf{f}}$) in the *nonlinear predictability* $\mathcal{P}_{\mathrm{N}}$ (Eq. 11), substituting Q' into $\mathcal{P}_{\mathrm{N}}$, and then replacing $\mathcal{P}_{\mathrm{L}}$ with $\mathcal{P}_{\mathrm{N}}$ in $\mathcal{O}_{\mathrm{L}}$ (Eq. 7) yields the following Rayleigh quotient-type objective to maximize:

$$\mathcal{O}_{\mathrm{N}}(\mathbf{f}) = \frac{\mathbf{f}^\top A \mathbf{f}}{\mathbf{f}^\top C_N \mathbf{f}}, \quad A = (C_N \mathbf{f}^t)(C_N \mathbf{f}^t)^\top + \lambda_{\mathrm{J}} Q'. \tag{13}$$

For any non-negative definite matrices A and C_N, the maximizer of the Rayleigh quotient $\mathcal{O}_{\mathrm{N}}$ is the largest eigenvector (the eigenvector corresponding to the maximum eigenvalue) of the generalized eigenvector problem $A\mathbf{f} = \lambda C_N \mathbf{f}$. The computational complexity of solving the generalized eigenvector problem of matrices $\{A, C_N\} \subset \mathbb{R}^{N\times N}$ is $O(N^3)$. As in our case $N = |X|$, solving this problem

is infeasible for large-scale problems. To obtain a computationally affordable solution, we first note that A incorporates multiplications by the centering matrix C_N and, therefore, all eigenvectors of A are centered, which implies that they are also eigenvectors of C_N. This effectively renders the generalized eigenvector problem into the standard eigenvector problem of matrix A.

Secondly, we make sparse approximate GP inference by adopting a low-rank approximation of K [20]:

$$K \approx K_{GB} K_{BB}^{-1} K_{GB}^\top,\ K_{GB[i,j]} = k(G_{[i,:]}, B_{[j,:]}),\ K_{BB[i,j]} = k(B_{[i,:]}, B_{[j,:]}), \quad (14)$$

where the i-th row $B_{[i,:]}$ of $B \in \mathbb{R}^{N' \times R}$ represents the i-th *basis vector*. We construct the basis vector matrix B by linearly sampling N' rows from all rows of G. Now substituting the kernel approximation in Eq. 14 into Eq. 12 leads to

$$Q'' = C_N K_{GB} (\lambda_J T) K_{GB}^\top C_N, \quad \text{with} \quad (15)$$

$$T = 2P - P K_{GB}^\top K_{GB} P \quad \text{and} \quad P = (K_{GB}^\top K_{GB} + \lambda K_{BB})^{-1}. \quad (16)$$

Replacing Q' in A with Q'', we obtain $A = YY^\top$, where

$$Y = \left[C_N \mathbf{f}^t, \sqrt{\lambda_J} K_{GB} T^{\frac{1}{2}} \right] \in \mathbb{R}^{N \times (N'+1)} \quad (17)$$

and $T^{\frac{1}{2}} (T^{\frac{1}{2}})^\top = T \in \mathbb{R}^{N' \times N'}$. Note that T is positive definite (PD) for $\sigma^2 > 0$ as Q'' is PD, which can be seen by noting that $0 \leq \frac{\mathbf{f}^\top Q'' \mathbf{f}}{\mathbf{f}^\top C_N \mathbf{f}} \leq 1$: by construction, $\mathbf{f}^\top Q'' \mathbf{f}$ is the prediction accuracy upper bounded by $\mathbf{f}^\top C_N \mathbf{f}$. Therefore, $T^{\frac{1}{2}}$ can be efficiently calculated based on the Cholesky decomposition of T. In the rare case where Cholesky decomposition cannot be calculated, e.g. due to round-off errors, we perform the (computationally more demanding) eigenvalue decomposition $E \Lambda E^\top$ of T, replace all eigenvalues in Λ that are smaller than a threshold $\varepsilon = 10^{-9}$ by ε, and construct $T^{\frac{1}{2}}$ as $E \Lambda^{\frac{1}{2}}$.

Finally, by noting that, when normalized, the largest eigenvector of $YY^\top \in \mathbb{R}^{N \times N}$ is the same as $Y\mathbf{e}$, where $\mathbf{e}$ is the largest eigenvector of $Y^\top Y \in \mathbb{R}^{(N'+1) \times (N'+1)}$, the optimum $\mathbf{f}^*$ of $\mathcal{O}_\mathrm{N}$ in Eq. 13 is obtained as $\frac{Y\mathbf{e}}{\|Y\mathbf{e}\|}$ and $\mathbf{e}$ can be efficiently calculated by iterating the power method on $Y^\top Y$. The normalized output $\mathbf{f}^*$ can be directly used in some applications, e.g. ranking. When the absolute values of predictors are important, e.g. in regression and multi-class classification, the standard deviation and the mean of $\mathbf{f}^0$ can be stored before the predictor combination process and $\mathbf{f}^*$ is subsequently inverse normalized.

3.3 Automatic identification of relevant tasks

Our algorithm *NPC* is designed to exploit all references. However, in general, not all references are relevant and therefore, the capability of identifying only relevant references can help. *OPC* does so by defining the weights $\{w_i\}$ (Eq. 1). However, this strategy inherits the limitation of *OPC* in that it does not consider all references jointly. An important advantage of our approach, formulating predictor combination as enhancing the predictability via Bayesian inference, is that the well-established methods of automatic relevance determination can be

employed for identifying relevant references. In our GP prediction framework, the contributions of references are controlled by the kernel function k (Eq. 10). The original Gaussian kernel k uses (isotropic) Euclidean distance $\|\cdot\|$ on $\mathcal{X}$ and thus treats all references equally. Now replacing it by an *anisotropic* kernel

$$k_A(\mathbf{a},\mathbf{b}) = \exp\left(-(\mathbf{a}-\mathbf{b})^\top \Sigma_A (\mathbf{a}-\mathbf{b})\right) \tag{18}$$

with $\Sigma_A = \mathrm{diag}\left[\sigma_A^1,\ldots,\sigma_A^R\right]$ being a diagonal matrix of non-negative entries renders the problem of identifying relevant references into estimating the hyperparameter matrix Σ_A: when σ_A^i is large, then $\mathbf{g}_i$ is considered relevant and it makes a significant contribution in predicting $\mathbf{f}$, while a small σ_A^i indicates that $\mathbf{g}_i$ makes a minor contribution.

For a fixed target predictor $\mathbf{f}$, identifying the optimal parameter Σ_A^* is a well-studied problem in Bayesian inference: Σ_A^* can be determined by maximizing the *marginal likelihood* [19] $p(\mathbf{f}\,|\,G,\Sigma_A)$. This strategy cannot be directly applied to our algorithm as $\mathbf{f}$ is the variable that is optimized depending on the prediction made by GPs. Instead, one could estimate Σ_A^* based on the initial prediction $\mathbf{f}^0$ and G, and fix it throughout the optimization of $\mathbf{f}$. We observed in our preliminary experiments that this strategy indeed led to noticeable performance improvement over using the isotropic kernel k. However, optimizing the GP marginal likelihood $P(\mathbf{f}\,|\,G,\Sigma_A)$ for a (nonlinear) Gaussian kernel (Eq. 10) is computationally demanding: this process takes roughly 1,000 times longer than the optimization of $\mathcal{O}_\mathrm{N}$ (Eq. 13; for the *AWA2* dataset case; see Sec. 4). Instead, we first efficiently determine surrogate parameters $\Sigma_L = \mathrm{diag}\left[\sigma_L^1,\ldots,\sigma_L^R\right]$ by optimizing the marginal likelihood based on the linear anisotropic kernel $k_L(\mathbf{a},\mathbf{b}) = \mathbf{a}^\top \Sigma_L \mathbf{b}$. In our preliminary experiments, we observed that once optimized, the relative magnitudes of Σ_L^* elements are similar to these of Σ_A^*, but their global scales differ (see the supplemental document for examples and details of marginal likelihood optimization). In our final algorithm, we determine Σ_A^* by scaling Σ_L^*: $\Sigma_A^* = \Sigma_L^*/\sigma_k^2$ for a hyperparameter $\sigma_k^2 > 0$.

Figure 2 demonstrates the effectiveness of automatic relevance determination: The *OSR* dataset contains 6 target attributes for each data instance, which are defined based on the underlying class labels. The figure shows the average diagonal values of Σ_L^* on this dataset estimated for the first attribute using the remaining 5 attributes, plus 8 additional attributes as references. Two scenarios are considered. In the *random references* scenario, the additional attributes are randomly generated. As indicated by small magnitudes and the corresponding standard deviations of Σ_L^* entries, our algorithm successfully disregarded these irrelevant references. In *class references* scenario, the additional attributes are ground-truth class labels which provide complete information about the target attributes. Our algorithm successfully *picks up* these important references. On average, removing the automatic relevance determination from our algorithm decreases the accuracy improvement (from the initial predictors $\mathbf{f}^0$) by 11.97% (see Table 2).

3.4 Joint denoising

So far, we assumed that all references in $\mathcal{G}$ are noise-free. However, in practice, they might be noisy estimates of the ground truth. In this case, noise in the references

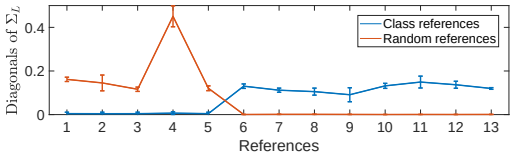

Fig. 2. The average diagonal values of Σ_L^* optimized for the first attribute of the *OSR* dataset as the target with remaining 5 attributes in the same dataset as references 1 to 5, plus 8 additional attributes as references 6 to 13. Σ_L^* values are normalized to sum to one for visualization. The length of each error bar corresponds to twice the standard deviation. *Class references*: References 6–13 are class labels from which attribute labels are generated. *Random references*: References 6–13 are randomly generated. See text.

Table 2. Effect of design choices in our algorithm on the *OSR* dataset. The average rank accuracy improvement over multiple target attributes from the baseline initial predictions $\mathbf{f}^0$ are shown (see Sec. 4 for details). *w/o joint denois.* only denoising the target predictor. *w/o auto. relev.*: without automatic relevance determination. Numbers in parentheses are accuracy ratios w.r.t. *Final NPC*.

Design choices →	*w/o joint denois.*	*w/o auto. relev.*	*Final NPC*
Accuracy improvement	1.96 (91.74%)	1.88 (88.03%)	**2.13 (100%)**

could be propagated to the target predictor during denoising, which would degrade the final output. We account for this by denoising *all* predictors $\mathcal{H}=\{\mathbf{f},\mathbf{g}_1,...,\mathbf{g}_R\}$ simultaneously. At each iteration t, each predictor $\mathbf{h}\in\mathcal{H}$ is denoised by considering it as the target predictor, and $\mathcal{H}\setminus\{\mathbf{h}\}$ as the references in Eq. 13. In the experiments with the *OSR* dataset, removing this joint denoising process from our final algorithm decreases the average accuracy rate by 8.26% (see Table 2). We provide a summary of our complete algorithm in the supplemental document.

Computational complexity and discussion. Assuming that $N \gg R$, the computational complexity of our algorithm (Eq. 13) is dominated by calculating the kernel matrix K_{GB} (Eq. 14), which takes $O(NN'R)$ for N data points, N' basis vectors and R references. The second-most demanding part is the calculation of $T^{\frac{1}{2}}$ from T based on Cholesky decomposition (Eq. 15; $O(N'^3)$). As we denoise not only the target predictor but also all references, the overall computational complexity of each denoising step is $O(R\times(NN'R+N'^3))$. On a machine with an Intel Core i7 9700K CPU and an NVIDIA GeForce RTX 2080 Ti GPU, the entire denoising process, including optimization of $\{(\Sigma_A)_i\}_{i=1}^R$ (Eq. 18), took around 10 seconds for the *AWA2* dataset with 37,322 data points and 79 references for each target attribute. For simplicity, we use the low-rank approximation of K (Eq. 14) for constructing sparse GP predictions, while more advanced methods exist [19]. The number N' of basis vectors is fixed at 300 throughout our experiments. While the accuracy of low-rank approximation (Eq. 14) is in general positively correlated with N', we have not observed any significant performance gain by raising N' to 1,000 in our experiments. GP predictions also generally improve

when *optimizing* the basis matrix B, e.g. via the marginal likelihood [21] instead of being selected from datasets as we did. Our efficient eigenvector calculation approach (Eq. 17) can still be applied in these cases.

4 Experiments

We assessed the effectiveness of our predictor combination algorithm in two scenarios: 1) visual attribute ranking [17], and 2) multi-class classification guided by the estimated visual attribute ranks. Given a database of images $X \subset \mathcal{X}$, visual attribute ranking aims to introduce a linear ordering of entries in X based on the strength of semantic attributes present in each image $\mathbf{x} \in X$. For a visual attribute, our goal is to estimate a rank predictor $f\colon \mathcal{X} \to \mathbb{R}$, such that $f(\mathbf{x}_i) > f(\mathbf{x}_j)$ when the attribute is stronger in $\mathbf{x}_i$ than $\mathbf{x}_j$. Parikh and Grauman's original relative attributes algorithm [17] estimates a linear rank predictor $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ via rank SVMs that use the rank loss $\mathcal{L}$ defined on ground-truth ranked pairs $U \subset X \times X$:

$$\mathcal{E}(f) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in U} \mathcal{L}(f, (\mathbf{x}_i, \mathbf{x}_j)) + C\|\mathbf{w}\|^2, \tag{19}$$

$$\mathcal{L}(f, (\mathbf{a}, \mathbf{b})) = \max(1 - (f(\mathbf{a}) - f(\mathbf{b})), 0)^2. \tag{20}$$

Yang et al. [24] and Meng et al. [14] extended this initial work using deep neural networks (*neural rankers*). Kim and Chang [10] extended the original predictor combination framework of Kim et al. [11] to rank predictor combination.

Experimental settings. For visual attribute ranking, we use seven datasets, each with annotations for multiple attributes per image. For each attribute, we construct an initial predictor and denoise it via predictor combination using the predictors constructed for the remaining attributes as the reference. The initial predictors are constructed by first training 1) neural rankers, 2) linear and 3) non-linear rank SVMs, and 4) semi-supervised rankers that use the iterated graph Laplacian-based regularizer [27], all using the rank loss $\mathcal{L}$ (Eq. 20). For each attribute, we select the ranker with the highest validation accuracy as *baseline* $\mathbf{f}^0 = f|_X$.

We compare our proposed algorithm to: 1) the baseline predictor $\mathbf{f}^0$, 2) Kim and Chang's adaptation [10] of Kim et al.'s predictor combination approach [11] to visual attribute ranking (*OPC*), and 3) Mejjati et al.'s multi-task learning (*MTL*) algorithm [13]. While the latter was originally designed for MTL problems, it does not require known forms of individual predictors and can be thus adapted for predictor combination. In the supplemental document, we also compare with an adaptation of Evgeniou et al.'s graph Laplacian-based MTL algorithm [4] to the predictor combination setting, which demonstrates that all predictor combination algorithms outperform naïve adaptations of traditional MTL algorithms.

Adopting the experimental settings of Kim et al. [10,11], we tune the hyper-parameters of all algorithms on evenly-split training and validation sets. Our algorithm requires tuning the noise level σ^2 (Eq. 12), global kernel scaling σ_k^2, and the regularization parameter λ_{J} (Eq. 13), which are tuned based on validation accuracy. For the number of iterations S, we use 20 iterations and select the

iteration number that achieves the highest validation accuracy. The hyperparameters for other algorithms are tuned similarly (see the supplemental material for details). For each dataset, we repeated experiments 10 times with different training, validation, and test set splits and report the average accuracies.

The *OSR* [17], *Pubfig* [17], and *Shoes* [12] datasets provide 2688, 772 and 14,658 images each and include rank annotations (i.e. strengths of attributes present in images) for 6, 11 and 10 visual attributes, respectively. The attribute annotations in these datasets were obtained from the underlying class labels. For example, each image in *OSR* is also provided with a ground-truth class label out of 8 classes. The attribute ranking is assigned per class-wise comparisons such that all images in a class have stronger (or the same) presence of an attribute than another class. This implies that the class label assigned for each image completely determines its attributes, while attributes themselves might not provide sufficient information to determine classes. Similarly, the attribute annotations for *Pubfig* and *Shoes* are generated from class labels out of 8 and 10 classes, respectively. The input images in *OSR* and *Shoes* are represented as combinations of GIST [15] and color histogram features, while *Pubfig* uses GIST features as provided by the authors [12,17]. In addition, for *OSR*, we extracted 2,048-dimensional features using ResNet101 pre-trained on ImageNet [7] to fairly assess the predictor combination performance when the accuracies of the initial predictors are higher thanks to advanced features (*OSR (ResNet)*).

The *aPascal* dataset is constructed based on the Pascal VOC 2008 dataset [3] containing 12,695 images with 64 attributes [5]. Each image is represented as a 9,751-dimensional feature vector combining histograms of local texture, HOG, and edge and color descriptors. The Caltech-UCSD Birds-200-2011 (*CUB*) dataset [22] provides 11,788 images with 312 attributes where the images are represented by the ResNet101 features. The Animals With Attributes 2 (*AWA2*) dataset consists of 37,322 images with 85 attributes [23]. We used the ResNet101 features as shared by Xian et al. [23]. For *aPascal*, *CUB*, and *AWA2*, the distributions of attribute values are imbalanced. To ensure that sufficient numbers (300) of training and testing labels exist for each attribute level, we selected 29, 40 and 80 attributes from *aPascal*, *CUB* and *AWA2*, respectively. The ranking accuracy is measured in $100\times$ Kendall's rank correlation coefficient, which is defined as the difference between the numbers of correctly and incorrectly ordered rank pairs, respectively, normalized by the number of total pairs (bounded in $100\times[-1,1]$; higher is the better).

The UT Zappos50K (*Zap50K*) contains 50,025 images of shoes with 4 attributes. Each image is represented as a combination of GIST and color histogram features provided by Yu and Grauman [25]. The ground-truth attribute labels are collected by instance-level pairwise comparison collected via *Mechanical Turk* [25].

We also performed multi-class classification experiments on the *OSR*, *Pubfig*, *Shoes*, *aPascal*, and *CUB* datasets based on their respective class labels. The initial predictors $\mathbf{f}^0 : \mathcal{X} \rightarrow \mathbb{R}^H$ are obtained as deep neural networks with continuous softmax decisions trained and validated on 20 labels per class. Each prediction is given as an H-dimensional vector with H being the number of classes. Our goal is to improve $\mathbf{f}^0$ using the predictors for visual attribute ranking as references. It

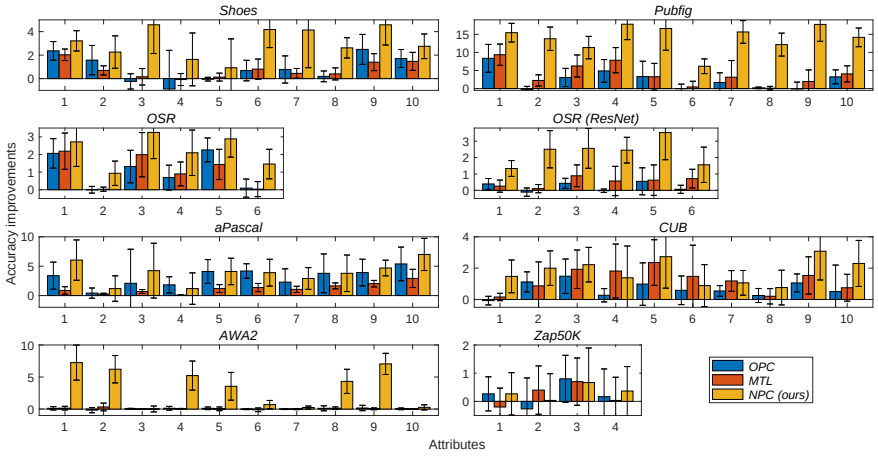

Fig. 3. Average accuracy improvement of different predictor combination algorithms from the *baseline predictors* for up to first 10 attributes. The complete results including statistical significance tests can be found in the supplemental document.

Table 3. Average classification accuracies (%) using rank estimates as references. The numbers in parentheses show the relative accuracy improvement over the baseline $\mathbf{f}^0$.

	Shoes	*Pubfig*	*OSR*	*aPascal*	*CUB*
Baseline $\mathbf{f}^0$	57.90 (0.00)	77.99 (0.00)	76.88 (0.00)	37.86 (0.00)	66.98 (0.00)
OPC [10]	58.52 (1.07)	82.55 (5.85)	77.16 (0.38)	39.77 (5.04)	67.75 (1.15)
MTL [13]	59.51 (2.78)	80.16 (2.78)	77.40 (0.68)	38.38 (1.37)	67.95 (1.45)
NPC (ours)	**62.87** (8.58)	**86.51** (10.9)	**79.71** (3.69)	**40.34** (6.55)	**68.26** (1.92)

should be noted that our algorithm *jointly improves* all H class-wise predictors as well as ranking references: 1) all class predictors evolve simultaneously, and 2) for improving the predictor of a class, the (evolving) predictors of the remaining classes are used as additional references. For a fair comparison, we denoise class-wise predictors using both the rank predictors and the predictors of the remaining classes as references, also for the other predictor combination algorithms.

Ranking results. Figure 3 summarizes the results for the relative attributes ranking experiments. Here, we show the results of only the first 10 attributes; the supplemental document contains complete results, which show a similar tendency as presented here. All three predictor combination algorithms frequently achieved significant performance gains over the baseline predictors $\mathbf{f}^0$. Importantly, apart from one case (*Shoes* attribute 4), all predictor combination algorithms did not significantly degrade the performance from the baseline. This demonstrates the utility of predictor combination. However, both *OPC* and *MTL* are limited in that they can only capture pairwise dependence between the target predictor and each reference. By taking into account the dependence present among the references, and thereby *jointly* exploiting them in improving the target predictor,

our algorithm further significantly improves the performance: Our algorithm performs best for 87.1% of attributes. In particular, *Ours* showed significant improvement on 6 out of 10 *AWA2* attributes, where the other algorithms achieved no noticeable performance gain. This supports our assumption that multiple attributes indeed can *jointly* supply relevant information for improving target predictors, even if not individually.

Multi-class classification results. Table 3 shows the results of improving multi-class classifications. Jointly capturing all rank predictors as well as the multi-dimensional classification predictions as references, our algorithm demonstrates significant performance gains (especially on *Shoes* and *Pubfig*), while other predictor combination algorithms achieved only marginal improvements, confirming the effectiveness of our joint prediction strategy.

5 Discussions and Conclusions

Our algorithm builds upon the assumption that the reference predictors can help improve the target predictor when they can well predict (or explain) the ground-truth $\mathbf{f}_{\mathrm{GT}}$. Since $\mathbf{f}_{\mathrm{GT}}$ is not available during testing, we use the noisy target predictor $\mathbf{f}^t$ at each time step t as a surrogate, which by itself is iteratively denoised. While our experiments demonstrate the effectiveness of this approach in real-world examples, simple failure cases exist. For example, if $\mathbf{f}^0$ (as the initial surrogate to $\mathbf{f}_{\mathrm{GT}}$) is contained in the reference set $\mathcal{G}$, our automatic reference determination approach will pick this up as the single most relevant reference, and therefore, the resulting predictor combination process will simply output $\mathbf{f}^0$ as the final result. We further empirically observed that even when the automatic relevance determination is disabled (i.e. $\Sigma_L = I$), the performance degraded significantly when $\mathbf{f}^0$ is included in $\mathcal{G}$. Also, as shown for the *Zap50K* results, there might be cases where no algorithm shows any significant improvement (indicated by the relatively large error bars). In general, our algorithm may fail when the references do not communicate sufficient *information* for improving the target predictor. Quantifying such utility of references and predicting the failure cases may require a new theoretical analysis framework.

Existing predictor combination algorithms only consider pairwise relationships between the target predictor and each reference. This misses potentially relevant information present in the dependence among the references. We explicitly address this limitation by introducing a new *predictability criterion* that measures how references are *jointly* contributing in predicting the target predictor. Adopting a fully Bayesian framework, our algorithm can automatically select informative references among many potentially irrelevant predictors. Experiments on seven datasets demonstrated the effectiveness of the proposed predictor combination algorithm.

Acknowledgements. This work was supported by UNIST's 2020 Research Fund (1.200033.01), National Research Foundation of Korea (NRF) grant NRF-2019-R1F1A1061603, and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.20200013360011001, Artificial Intelligence Graduate School support (UNIST)) funded by the Korean government (MSIT).

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning **73**(3) (2008) 2, 3
2. Chen, L., Zhang, Q., Li, B.: Predicting multiple attributes via relative multi-task learning. In: CVPR. pp. 1027–1034 (2014) 2, 3
3. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. IJCV **111**(1), 98–136 (2015) 12
4. Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. JMLR **6**, 615–637 (2005) 11
5. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. pp. 1778–1785 (2009) 12
6. Gong, P., Ye, J., Zhang, C.: Robust multi-task feature learning. In: KDD. pp. 895–903 (2012) 2, 3, 4
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1, 12
8. Hein, M., Maier, M.: Manifold denoising. In: NIPS. pp. 561–568 (2007) 3
9. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD. pp. 133–142 (2002) 1
10. Kim, K.I., Chang, H.J.: Joint manifold diffusion for combining predictions on decoupled observations. In: CVPR. pp. 7549–7557 (2019) 3, 5, 11, 13
11. Kim, K.I., Tompkin, J., Richardt, C.: Predictor combination at test time. In: ICCV. pp. 3553–3561 (2017) 1, 2, 3, 5, 11
12. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Image search with relative attribute feedback. In: CVPR. pp. 2973–2980 (2012) 12
13. Mejjati, Y.A., Cosker, D., Kim, K.I.: Multi-task learning by maximizing statistical dependence. In: CVPR. pp. 3465–3473 (2018) 3, 11, 13
14. Meng, Z., Adluru, N., Kim, H.J., Fung, G., Singh, V.: Efficient relative attribute learning using graph neural networks. In: ECCV. pp. 552–567 (2018) 11
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV **42**(3), 145–175 (2001) 12
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010) 2
17. Parikh, D., Grauman, K.: Relative attributes. In: ICCV. pp. 503–510 (2011) 11, 12
18. Passos, A., Rai, P., Wainer, J., Daumé III, H.: Flexible modeling of latent task structures in multitask learning. In: ICML. pp. 1103–1110 (2012) 2, 3, 4
19. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006) 7, 9, 10
20. Seeger, M., Williams, C.K.I., Lawrence, N.D.: Fast forward selection to speed up sparse Gaussian process regression. In: International Workshop on Artificial Intelligence and Statistics (2003) 8
21. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: NIPS (2006) 11
22. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 12
23. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning – A comprehensive evaluation of the good, the bad and the ugly. IEEE TPAMI **41**(9), 2251–2265 (2019) 12

24. Yang, X., Zhang, T., Xu, C., Yan, S., Hossain, M.S., Ghoneim, A.: Deep relative attributes. IEEE T-MM **18**(9), 1832–1842 (2016) 11
25. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: CVPR. pp. 192–199 (2014) 12
26. Zamir, A.R., Sax, A., Shen, W., Guibas, L., Malik, J., Savarese, S.: Taskonomy: disentangling task transfer learning. In: CVPR. pp. 3712–3722 (2018) 2, 3, 4
27. Zhou, X., Belkin, M., Srebro, N.: An iterated graph Laplacian approach for ranking on manifolds. In: KDD. pp. 877–885 (2011) 11