Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-target regression via non-linear output structure learning

Shervin Rahimzadeh Arashloo [a,*], Josef Kittler [b]

[a] Department of Computer Engineering, Faculty of Engineering, Bilkent University, Ankara, Turkey
[b] Centre for Vision, Speech, and Signal Processing, University of Surrey, Guildford, UK

ARTICLE INFO

Article history:
Received 18 June 2021
Revised 9 October 2021
Accepted 12 December 2021
Available online xxxx

Keywords:
Multi-output regression
Non-linear structure learning
Vector-valued functions in the reproducing kernel Hilbert space (RKHSvv)
Tikhonov regularisation

ABSTRACT

The problem of simultaneously predicting multiple real-valued outputs using a shared set of input variables is known as multi-target regression and has attracted considerable interest in the past couple of years. The dominant approach in the literature for multi-target regression is to capture the dependencies between the outputs through a linear model and express it as an output mixing matrix. This modelling formalism, however, is too simplistic in real-world problems where the output variables are related to one another in a more complex and non-linear fashion. To address this problem, in this study, we propose a structural modelling approach where the correlations between output variables are modelled using a non-linear approach. In particular, we pose the multi-target regression problem as one of vector-valued composition function learning in the reproducing kernel Hilbert space and propose a non-linear structure learning approach to capture the relationship between the outputs via an output kernel. By virtue of using a non-linear output kernel function, the proposed approach can better discover non-linear dependencies among targets for improved prediction performance. An extensive evaluation conducted on different databases reveals the benefits of the proposed multi-target regression technique against the baseline and the state-of-the-art methods.

1. Introduction

Multi-target regression has received considerable attention due to its widespread use in many application domains including ecological modelling [1], economics [2], network datasets [3], natural language processing [4], computer vision [5,6], bioinformatics [7], education [8], marketing [9], signal processing [10], signal denoising and enhancement [11], etc. The major challenges in multi-target regression are mainly due to the following two aspects of the problem: 1) capturing the relationship between the input variables and the outputs; and 2) modelling and exploiting inter-output dependencies to enhance the predictive capability. The former task was conventionally addressed via a linear model to link the inputs to each output. However, due to the limited expressive capacity of a linear model, non-linear approaches for input–output modelling based on, for instance, non-linear kernel machines have been proposed [12,13]. Regarding the second task, it is known that in the presence of correlations between output variables, utilising the shared information across multiple relevant targets through a joint modelling mechanism provides performance advantageous compared to the case where each target variable is modelled independently [14,15,13,12,16,17]. Specifically, sharing knowledge among several tasks via exploiting the similarities between different problems may improve the generalisation capability of each learner, and decrease the number of observations needed for training, as well as the number of iterations to achieve a specific level of performance. Accordingly, multi-target learning is identified as a compelling technique of inductive transfer, that improves generalisation by making use of domain specific information inherent in the training samples of several tasks as an inductive bias [18]. This goal is very often achieved through a simultaneous learning of several problems, while utilising a shared input representation. For a multi-task learning, if a known structure exists between target variables, it may be directly deployed to enhance the prediction performance. Nevertheless, in many real-world learning problems, the dependencies among output variables are not known in advance. This necessitates the design of learning mechanisms to capture and model any possible dependencies between outputs using the available training data. In this context, there has been a large body of research to model dependencies between target variables for multi-target regression. The current approaches presume a non-linear input–output relation but typically model the inter-target dependencies using linear models. As a result, in these methods, each output variable is formed as a linear mixture of all

* Corresponding author.
E-mail address: s.rahimzadeh@cs.bilkent.edu.tr (S. Rahimzadeh Arashloo).

https://doi.org/10.1016/j.neucom.2021.12.048
0925-2312/

intermediate target variables. We argue that despite the appealing properties of this modelling formalism including, for instance, simpler models and learning procedures, in many real-world applications, assuming a linear relationship between multiple output variables is neither realistic nor sufficient. A linear inter-target assumption limits the representational capacity of these methods by discarding non-linear dependencies between multiple targets. Very often in practice, different outputs correspond to higher level concepts that give rise to highly complex relationships that demand effective non-linear output structure learning machines.

In this study, we address the multi-target regression problem and propose an output structure learning approach that not only learns non-linear relationships between the inputs and the outputs but is also capable of capturing non-linear inter-target relations. For this purpose, we approach the multi-target regression problem in a principled way and pose it as one of learning a vector-valued composition function in the RKHSvv (Reproducing Kernel Hilbert Space for Vector-Valued functions) [19]. In the proposed method, an input kernel is responsible for relating the inputs to outputs while the output kernel captures inter-target dependencies. By choosing the output kernel as an admissible non-linear kernel, the proposed method learns a non-linear structure among target variables. Needless to say that, if desired, by selecting the output kernel as a linear kernel, the proposed method simplifies to a linear structure learning technique. From this perspective, our approach is a generalisation of the structure learning methods in the RKHSvv and can handle both linear and non-linear structure learning problems.

1.1. Our Contributions

The current study makes the following contributions:

- We cast multi-target regression as one of non-linear output structure learning in the context of composition functions in RKHSvv and formulate the learning problem as an optimisation task, encoding both data fidelity and regularisation;
- We propose an effective method for the optimisation of the cost function associated with the proposed approach with guaranteed convergence;
- We present an evaluation of the proposed multi-target regression technique on different databases and provide a comparison to other techniques.

1.2. Organisation of the Paper

The rest of this paper is structured as described next. In Section 2, a review of multi-target regression methods with a focus on the output kernel learning algorithms is presented. In Section 3, once a background on vector-valued functionals in the Hilbert space and multi-target regression is provided, we introduce our non-linear output structure learning approach. In Section 4, we describe an alternating optimisation approach to optimise the objective function associated with the proposed method. In Section 5, after introducing the datasets used in the experiments, the results of an evaluation of the proposed method along with a comparison to the state-of-the-art techniques from the literature on several datasets are discussed. Finally, Section 6 provides conclusions.

2. Prior Work

There exists a diverse set of different approaches developed for multi-target regression. For instance, in [20], the authors propose an approach for multiple-target regression that models the structure through covariance estimation of the hidden model parameters in addition to the conditional structure represented as covariance matrix of the observed targets. Other work [21], proposes a method to model the correlation between the output variables via a sparse modelling of a multi-output regression coefficient matrix. The method incorporates a penalised likelihood term and simultaneously estimates the covariance structure and the regression coefficients. In [22], a different approach is presented that adjusts the regularisation for each individual regression problem according to its noise level so that it simultaneously achieves enhanced finite-sample performance and insensitivity to tuning.

In [23], the so-called clustered multi-target learning (CMTL) method is presented, assuming that different problems can be grouped into clusters, and that the problems within each cluster possess similar weight vectors. A novel spectral norm is then introduced which captures this deductive assumption, without using the prior information regarding the cluster of problems into groups, yielding a convex optimisation problem for multi-task inference. In [24], a novel method for CMTL, dubbed flexible clustered multi-task (FCMTL), is proposed, where the group structure is inferred through identification of representative problems. In contrary to its counterpart, the proposed method possesses higher flexibility as it does not need disjoint clusters and the problems within each cluster do not need to perform an information sharing at the same level. Furthermore, the method automatically infers the number of problem clusters directly from data.

Ensemble-based approaches have been also proposed for the multi-target regression analysis. As an instance, the work in [25], presents the fitted rule ensembles (FIREs) method to enhance multi-output regression performance by including linear base learners into the ensemble. Nevertheless, the performance of this algorithm is slightly worse than that of the multi-objective random forests method [26]. In [27], a symbolic regression approach based on Gene Expression Programming is proposed for the multi-target regression problem. The method can estimate the inter-target correlations using some genetic operators. Moreover, three ensemble approaches are proposed to better utilise the inter-output and input–output relations.

In another work [28], the objective function for multi-task learning is represented in terms of a linear fusion of two groups of eigen-functions so that the eigen-functions of a problem supply extra information to the other problem and assist to enhance its performance. Other study in [29] develops a two-layer method to concurrently learn hidden features that are shared between different tasks and a multi-target approach drawing on the Gaussian process formalism. In [30], a novel prototype selection mechanism for multi-output regression data sets is proposed where a multi-objective evolutionary technique is deployed for prototype selection. Other work [31] proposes an aim-object driven neuro-fuzzy asymmetric multi-target regression technique and a hybrid learning method that fuses the whale optimization and the recursive least-square estimator to train the model.

In [15], for exploiting the solution of a suitable regularisation task in a RKHSvv, an output kernel learning approach is proposed. For optimisation, a block-wise coordinate descent approach is proposed which efficiently utilises the structure of the multiple problems. Other work [13], considers multi-target learning by showing that several problems and the structure between them may be effectively inferred by solving a convex optimisation problem using a block coordinate optimisation method. In a more recent study [12], the authors present a multi-output regression technique through learning low-rank matrices. Using matrix elastic nets, this method can capture inter-response dependencies in a structure matrix. In another study [16] a multi-target sparse latent regression model is proposed to capture inherent inter-output

dependencies and non-linear complex relationships between the inputs and the outputs. In this method, inter-target dependencies are modelled using $l_{2,1}$-norm-based sparse learning. A different study [17], uses the RKHSvv theory to account for the structure in the observations, while also utilising kernels in the input sample space.

Motivated by their outstanding modelling capability, deep neural networks have recently attracted attention for learning highly complex functions, in particular in the analysis of audio-visual data [32]. While such methods have proven to be effective for modelling non-linear functions, nevertheless, their applicability is typically limited to the problems where large amount of training data is available as such networks typically possess a large number of free parameters that should be tuned using the training samples. One strong alternative to the deep learning approaches is that of kernel-based methods [12,13]. In comparison to the deep learning methods, kernel-based approaches typically require less training data and are based on well understood mathematical principles. The proposed approach in this work falls into the kernel-based group of multi-target regression techniques. One may consult [33] for a more detailed review of the related work on the multi-target regression problem.

3. Methodology

In this section, once a brief introduction to the vector-valued functions in the Hilbert space and multi-output regression is provided, the proposed multi-target regression approach shall be introduced.

3.1. Hilbert Space Vector Functions

Let's suppose there exist M scalar learning problem, each of which is provided with a data set D_m of n_m input-response samples $D_m = \{(x_i^m, r_i^m)\}_{i=1}^{n_m}$ for training where $x_i^m \in \mathscr{X}$ stands for the input observation while $r_i^m \in \mathbb{R}$ corresponds to the output response, and $m \in \{1,\ldots,M\}$ indexes a specific problem. Considering an objective function $\mathscr{O} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ which gauges per-task prediction errors, the goal in inferring functions with vector values in a Hilbert space is to determine a functional $\mathbf{B}(.)$ that simultaneously optimises the errors associated with different learning tasks, i.e. $\mathbf{B}^*(.) = \arg\min_{\mathbf{B}\in\mathscr{H}} P$, where P is given as

$$P = \sum_{m=1}^{M} \frac{1}{n_m} \sum_{i=1}^{n_m} \mathscr{O}(r_i^m, b_m(x_i^m)) + \mathscr{R}(\mathbf{B}) \tag{1}$$

where $\mathscr{R}(\mathbf{B})$ enforces a regularisation in the Hilbert space on the functional $\mathbf{B}(.)$, whose scalar elements are b_m.

A sub-category of the problems in vector-valued function learning corresponds to kernel space multi-output regression where the loss functional $\mathscr{O}$ captures a sum of squared losses in the Hilbert space. In this formalism, a widely used simplifying hypothesis corresponds to the separability of input–output relationships, that results in expressing $\mathbf{B}(.)$ through a kernel which is a separable function. Separable functions correspond to kernels expressible in the form of $\mathbf{L}(x_1, x_2) = \kappa(x_1, x_2)\mathbf{\Pi}$, where $\mathbf{\Pi}$ denotes a symmetric $M \times M$ matrix, which is positive semi-definite, and captures the correlations between the outputs while κ denotes a reproducing kernel that is a scalar function with the domain of $\mathscr{X} \times \mathscr{X}$ and a range of $\mathbb{R}$. $\kappa(.,.)$ encodes similarities among the input samples. Function $\mathbf{B}(.)$ can then be expressed as

$$\mathbf{B}(.) = \sum_{i=1}^{n} \kappa(x_i, .)\mathbf{\Pi}\boldsymbol{\lambda}_i \tag{2}$$

where n is the total number of training instances from all tasks, i.e. $n = \sum_{m=1}^{M} n_m$ and $\boldsymbol{\lambda}_i$ represents the coefficients. In this case, using matrix notation, the corresponding outputs for the training samples may be obtained as $\mathbf{K\Lambda\Pi}$, and hence, the regularised objective function in Eq. 1 can be written as

$$P = \|\mathbf{K\Lambda\Pi} - \mathbf{R}\|_2^2 + \mathscr{R}(\mathbf{K}, \mathbf{\Lambda}, \mathbf{\Pi}) \tag{3}$$

where $\mathbf{K}^{n\times n}$ stands for the inputs kernel matrix while $\mathbf{\Lambda}^{n\times M}$ denotes the coefficients matrix, $\mathbf{R}$ represents a matrix of the true outputs and $\|.\|_2^2$ stands for the matrix Frobenius norm. For the class of separable kernels, when $\mathbf{\Pi}$ corresponds to an identity matrix, the responses for all observations shall be regarded as independent and the multi-target learning task will simplify to solving each individual problem independently. Under the condition that the structure matrix $\mathbf{\Pi}$ differs from the identity matrix, different problems are considered as being correlated and solving for the optimal functional $\mathbf{B}(.)$ is formulated as the problem of learning $\mathbf{\Lambda}$ and $\mathbf{\Pi}$, simultaneously, subject to some regularisation constraints. The general functional form of the objective P in Eq. 3 can be assumed as the widely used formulation of the multi-output regression task in the literature, where regularisation choices for $\mathscr{R}$ may be driven by distinct a priori assumptions, giving rise to different instances of the task. Considering the separable kernel exposition of the multi-output regression problem, the responses of a multi-output method may be regarded as computing the intermediate outputs for each separate task using $\mathbf{K\Lambda}$ which are then mixed together using a dependency encoding approach to generate the outputs. From this viewpoint, the outputs can be regarded as the responses of a composition functional $\mathbf{B}(.) = \mathbf{h}(\mathbf{g}(.))$, where $\mathbf{g}(.)$ generates intermediate outputs, whereas $\mathbf{h}(.)$ applies a mixing of the intermedial outputs to produce the responses. In this context, the relation in Eq. 2 represents a non-linear projection functional $\mathbf{g}(.)$ that is formulated in terms of a non-linear kernel $\kappa(.,.)$ and the relevant coefficients $\mathbf{\Lambda}$, while function $\mathbf{h}(.)$ is expressed as a linear fusion function, specified via $\mathbf{\Pi}$. Most of the existing methods for the multi-target regression problem is concentrated on modelling $\mathbf{h}(.)$ as a linear function.

3.2. The proposed approach

In this study, we address the problem of simultaneously learning multiple regression tasks, where each problem is characterised as a kernel regression, modelling the individual predictions in terms of the elements of a vector functional. The loss function proposed may be represented as

$$P = \|\mathbf{h}(\mathbf{K\Lambda}) - \mathbf{R}\|_2^2 + \mathscr{R}(\mathbf{K}, \mathbf{\Lambda}, \mathbf{\Pi}) \tag{4}$$

In contrast to the existing approaches, we assume $\mathbf{h}(.)$ to be a non-linear (kernel) composition function to capture the non-linear relational structure of multiple regression problems and draw on a generic Representer theorem for composition functions in the reproducing kernel Hilbert space given in [34]. The proposed structure of the learning machine is depicted in Fig. 1. In the proposed approach, after producing the intermediate responses of different tasks (y_m's, for $m = 1,\ldots,M$) for a data point $\mathbf{x}$, the generated responses as a whole are considered as one vectorial sample (i.e. $\mathbf{y}$) and fed to the second layer. In the next stage, the generated response vector by the first layer, i.e. $\mathbf{y}$, is non-linearly projected onto the space induced by a Gaussian kernel and finally fused using $\mathbf{\Pi}$ to generate the ultimate outputs for each regression task. As such, the training/test data samples for the second layer correspond to M-dimensional intermediate outputs collected into vectors where M stands for the number of regression tasks. Assuming $\kappa_1(.,.)$ and

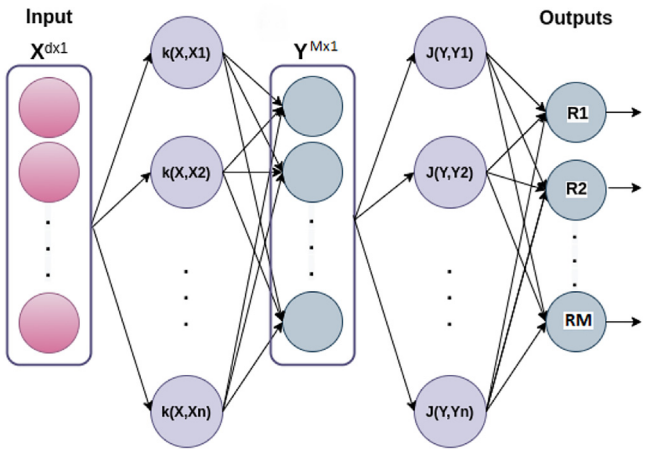

Fig. 1. The proposed Non-linear Output Structure Learning (NOSL) machine for multi-target regression.

$\kappa_2(.,.)$ as the kernel functions associated with the first (the one closer to the input) and the second layers respectively, the function $\mathbf{B}(.)$ in the proposed non-linear structure learning approach can be represented as

$$\mathbf{B}(.) = \sum_{i=1}^{n} \kappa_2 \left(\sum_{j=1}^{n} \kappa_1(x_j, .)\boldsymbol{\lambda}_j, \sum_{j=1}^{n} \kappa_1(x_j, x_i)\boldsymbol{\lambda}_j \right) \boldsymbol{\Pi}_i \tag{5}$$

where $\boldsymbol{\Pi}_i$ denotes the transpose of the i^{th} row of the second-layer coefficient matrix $\boldsymbol{\Pi}^{n\times M}$ while the coefficients of the first layer are the M-element vectors $\boldsymbol{\lambda}_j$'s. The second-layer kernel matrix $\mathbf{J}$ is constructed using $\kappa_2(.,.)$ fed with the intermediate outputs $\mathbf{Y} = \mathbf{K}\boldsymbol{\Lambda}$ where $\boldsymbol{\Lambda}$ denotes a matrix collection of $\boldsymbol{\lambda}_j$'s and $\mathbf{K}$ represents the kernel matrix associated with the first layer which is built using $\kappa_1(.,.)$. Using a Gaussian kernel function for the second layer, i.e. $\kappa_2(y_i, y_j) = \exp(-\theta y_i - y_j)$, we have

$$\mathbf{J} = \exp\left(-\theta\left[(\mathbf{I}\odot\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K})\mathbf{1} + \mathbf{1}^\top(\mathbf{I}\odot\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K})^\top - 2\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K}\right]\right) \tag{6}$$

where $\odot$ denotes the Hadamard product while $\mathbf{1}$ represents a matrix of 1's. The RBF kernel width in the second layer is controlled by the scalar parameter θ. Based on the definition of $\mathbf{B}(.)$ in Eq. 5 and using the relation for $\mathbf{J}$ in Eq. 6, using a matrix notation the responses over all training observations are given as $\mathbf{J}\boldsymbol{\Pi}$. In other words, the nonlinear function $\mathbf{h}(\mathbf{K}\boldsymbol{\Lambda})$ in Eq. 4 for the proposed approach may be explicitly expressed as

$$\mathbf{h}(\mathbf{K}\boldsymbol{\Lambda}) = \exp\left(-\theta\left[(\mathbf{I}\odot\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K})\mathbf{1} + \mathbf{1}^\top(\mathbf{I}\odot\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K})^\top - 2\mathbf{K}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top\mathbf{K}\right]\right)\boldsymbol{\Pi} \tag{7}$$

In the proposed approach for learning the non-linear structures between multiple regression problems, the unknown first-layer coefficient matrix $\boldsymbol{\Lambda}$ and the second-layer coefficient matrix $\boldsymbol{\Pi}$ are determined through optimising the loss function P that corresponds to a regularised kernel regression defined in terms of $\mathbf{J}$. The merits of the proposed non-linear structure learning approach, when compared to the commonly used linear structure learning approaches, can be justified from the viewpoint that the conventional linear structure operates as a linear regressor on the intermediate outputs, while the proposed method depicted in Fig. 1 represents a non-linear kernel regression.

3.3. Regularisation

Tikhonov is a widely used regularisation scheme in the context of regularised regression. A Tikhonov regularisation in the non-linear multi-target formulation promotes coefficients that generate outputs, which are produced from the intermediate responses through smooth functions by imposing a penalty on larger magnitude parameters and therefore generating a more parsimonious coefficient. For a Tikhonov regularisation, the objective for the proposed model depicted in Fig. 1, is given as

$$P = \|\mathbf{J}\boldsymbol{\Pi} - \mathbf{R}\|_2^2 + \gamma_1 \text{trace}(\boldsymbol{\Lambda}^\top\mathbf{K}\boldsymbol{\Lambda}) + \gamma_2 \text{trace}(\boldsymbol{\Pi}^\top\mathbf{J}\boldsymbol{\Pi}) \tag{8}$$

where $\mathbf{J}$ and $\mathbf{K}$ represent the second- and the first-layer kernel matrices, respectively. Note that the trace(.) operator imposes a Tikhonov regularisation on the coefficients of the first and the second layers. The data fidelity term, as discussed above, corresponds to a kernel regression loss function (sum of squared errors) fed with the responses generated by the first layer.

3.4. Relation to Stacking Ensembles

In the stacking ensemble methods [35], once a number of first-level learners are trained, a second-layer learner is trained on the predictions of the base learners to combine and improve on the prediction performance of the base learners. From this perspective, the proposed approach (and also all output kernel learning methods) operates in a similar fashion, as the predictions of first-layer regressors are combined by the second-layer learner. However, there exists a subtle difference between such stacking ensemble techniques and the proposed method. In the stacking ensemble framework, the second-layer learner only combines the first-layer predictions. If there exists a prediction error, the base learners have no means of being informed and updated to correct for the wrong prediction. However, in the proposed method, due to the optimisation that involves the parameters of both layers recursively, the second-layer learner not only combines the first-layer predictions but also provides an effective mechanism to adjust the parameters of the first-layer base learners, as discussed next.

4. Alternating Optimisation

For minimising the objective function in the proposed approach, a block coordinate descent technique that alternates between optimising the first-layer and the second-layer parameters is used in this work, as explained in the next section.

4.1. Fix $\boldsymbol{\Pi}$ to Optimise $\boldsymbol{\Lambda}$

The first direction for the optimisation of the loss function P corresponds to $\boldsymbol{\Lambda}$. The partial derivatives of the first regularisation term, i.e. trace$(\boldsymbol{\Lambda}^\top\mathbf{K}\boldsymbol{\Lambda})$ with respect to $\boldsymbol{\Lambda}$ are readily derived as

$$\frac{\partial \text{trace}(\boldsymbol{\Lambda}^\top\mathbf{K}\boldsymbol{\Lambda})}{\partial\boldsymbol{\Lambda}} = 2\mathbf{K}\boldsymbol{\Lambda} \tag{9}$$

By representing other terms in P as $P_1 = \|\mathbf{J}\boldsymbol{\Pi} - \mathbf{R}\|_2^2 + \gamma_2 \text{trace}(\boldsymbol{\Pi}^\top\mathbf{J}\boldsymbol{\Pi})$, we will compute its partial derivative with respect to $\boldsymbol{\Lambda}$. None of the terms in P_1 depend on $\boldsymbol{\Lambda}$, except the second-layer kernel matrix $\mathbf{J}$ (recall that $\mathbf{J}$ models the similarities between M-dimensional intermediate responses $\mathbf{y}$'s). To derive the partial derivatives of P_1 with respect to $\boldsymbol{\Lambda}$, let us first define the following matrices:

$$\mathbf{C} = \mathbf{Y}\mathbf{Y}^\top\mathbf{D} = (\mathbf{I}\odot\mathbf{C})\mathbf{1} + \mathbf{1}^\top(\mathbf{I}\odot\mathbf{C})^\top - 2\mathbf{C} \tag{10}$$

where $\odot$ denotes the Hadamard product while $\mathbf{1}$ represents a matrix of 1's. Using the definitions above, the second-layer kernel matrix $\mathbf{J}$ can be written as

$$\mathbf{J} = \exp[-\theta \mathbf{D}] \tag{11}$$

where the RBF kernel width in the second layer is controlled by the scalar parameter θ. The partial derivatives of P_1 with respect to the kernel matrix $\mathbf{J}$ are

$$\frac{\partial P_1}{\partial \mathbf{J}} = 2(\mathbf{J}\boldsymbol{\Pi} - \mathbf{R})\boldsymbol{\Pi}^\top + \gamma_2 \boldsymbol{\Pi}\boldsymbol{\Pi}^\top \tag{12}$$

The partial derivatives $\partial P_1/\partial \mathbf{D}, \partial P_1/\partial \mathbf{C}, \partial P_1/\partial \mathbf{Y}$ are

$$\begin{aligned} \tfrac{\partial P_1}{\partial \mathbf{D}} &= (-\theta \mathbf{J}) \odot \tfrac{\partial P_1}{\partial \mathbf{J}} \\ \tfrac{\partial P_1}{\partial \mathbf{C}} &= \mathbf{I}_n \odot \left(\left(\tfrac{\partial P_1}{\partial \mathbf{D}} + \tfrac{\partial P_1}{\partial \mathbf{D}}^\top \right) \mathbf{1}^\top \right) - 2\tfrac{\partial P_1}{\partial \mathbf{D}} \\ \tfrac{\partial P_1}{\partial \mathbf{Y}} &= \left(\tfrac{\partial P_1}{\partial \mathbf{C}} + \tfrac{\partial P_1}{\partial \mathbf{C}}^\top \right) \mathbf{Y} \end{aligned} \tag{13}$$

For the computation of $\partial P_1/\partial \boldsymbol{\Lambda}$ we note

$$\delta P_1 = \mathrm{trace}\left(\frac{\partial P_1}{\partial \mathbf{Y}}^\top \delta \mathbf{Y} \right) = \mathrm{trace}\left(\frac{\partial P_1}{\partial \boldsymbol{\Lambda}}^\top \delta \boldsymbol{\Lambda} \right) \tag{14}$$

Since $\mathbf{Y} = \mathbf{K}\boldsymbol{\Lambda}$, one obtains $\delta \mathbf{Y} = \mathbf{K}\delta\boldsymbol{\Lambda}$. Substituting $\delta \mathbf{Y}$ by $\mathbf{K}\delta\boldsymbol{\Lambda}$ in Eq. 14 gives

$$\delta P_1 = \mathrm{trace}\left(\frac{\partial P_1}{\partial \mathbf{Y}}^\top \mathbf{K}\delta \boldsymbol{\Lambda} \right) = \mathrm{trace}\left(\frac{\partial P_1}{\partial \boldsymbol{\Lambda}}^\top \delta \boldsymbol{\Lambda} \right) \tag{15}$$

and hence

$$\frac{\partial P_1}{\partial \boldsymbol{\Lambda}} = \mathbf{K}\frac{\partial P_1}{\partial \mathbf{Y}} \tag{16}$$

Summarising the procedure described above, for computing $\partial P_1/\partial \boldsymbol{\Lambda}$, first $\partial P_1/\partial \mathbf{J}$ should be computed followed by $\partial P_1/\partial \mathbf{D}, \partial P/\partial \mathbf{C}$ and $\partial P_1/\partial \mathbf{Y}$, respectively, and then $\partial P_1/\partial \boldsymbol{\Lambda}$. Ultimately, $\partial P/\partial \boldsymbol{\Lambda} = \partial P_1/\partial \boldsymbol{\Lambda} + 2\gamma_1 \mathbf{K}\boldsymbol{\Lambda}$.

4.2. Fix $\boldsymbol{\Lambda}$ to Optimise $\boldsymbol{\Pi}$

In order to optimise the regularised error over multiple problems, represented by P with respect to $\boldsymbol{\Pi}$, one can set its partial derivative $\partial P/\partial \boldsymbol{\Pi}$ to 0:

$$\frac{\partial P}{\partial \boldsymbol{\Pi}} = 2\mathbf{J}^\top(\mathbf{J}\boldsymbol{\Pi} - \mathbf{R}) + 2\gamma_2 \mathbf{J}\boldsymbol{\Pi} = 0 \tag{17}$$

which results in

$$\boldsymbol{\Pi} = (\mathbf{J} + \gamma_2 \mathbf{I}_n)^{-1}\mathbf{R} \tag{18}$$

Lastly, the partial derivatives of the cost function P w.r.t. θ are

$$\frac{\partial P}{\partial \theta} = \mathrm{trace}\left(\frac{\partial P}{\partial \mathbf{J}}^\top (-\mathbf{J} \odot \mathbf{D}) \right) \tag{19}$$

To minimise the objective function in θ and $\boldsymbol{\Lambda}$, a gradient descent approach may be utilised. The proposed method is summarised in Algorithm 1, where η_θ and η_A stand for the step sizes for θ and $\boldsymbol{\Lambda}$, in the gradient descent procedure, respectively. It is worth noting that in Algorithm 1, in Step 6, one must update the second-layer kernel matrix $\mathbf{J}$ using the most recent updated values for $\boldsymbol{\Lambda}$ and θ. In general, in all kernel-based methods utilising a Gaussian kernel, the kernel width parameter (θ) is required to be a non-negative scalar. Assuming $\theta \geqslant 0$, the two extreme values for $J = \exp(-\theta D)$ would be 0 and 1. In our experiments, we have used a line search to determine the step sizes to guarantee a decrease in the objective function value when updating θ. In these cases, θ always stayed positive, in spite of the fact that we did not explicitly enforce a zero lower bound for θ. In other words, using suitable gradient step sizes, the optimal value for θ was never negative. In a more general setting, however, one may impose an explicit non-negativity constraint on θ by projecting it onto the positive orthant to ensure the stability of the Gaussian term if required.

Algorithm 1: The Proposed Multi-Target Regression Method

1: $\boldsymbol{\Lambda} = (\mathbf{K} + \gamma_1 \mathbf{I}_n)^{-1}\mathbf{R}$
2: $\theta = 1/\mathrm{m}_{\mathbf{D}}$
3: **Repeat**
4: $\quad \boldsymbol{\Lambda} = \boldsymbol{\Lambda} - \eta_A \frac{\partial Q_N}{\partial \boldsymbol{\Lambda}}$
5: $\quad \theta = \theta - \eta_\theta \frac{\partial Q_N}{\partial \theta}$
6: $\quad \mathbf{J} = \mathbf{J}(\boldsymbol{\Lambda}, \theta)$
7: $\quad \boldsymbol{\Pi} = (\mathbf{J} + \gamma_2 \mathbf{I}_n)^{-1}\mathbf{R}$
8: **Until** $|P^{t+1} - P^t| < \zeta$

4.3. Initialisation

During the initialisation step of the proposed multi-target approach, parameter θ that tunes the RBF kernel width in the second layer of the proposed structure learning approach is set to 1 over the average Euclidean distance between all training observations, i.e. $\theta = 1/m_{\mathbf{D}}$ where $m_{\mathbf{D}}$ stands for the mean of $\mathbf{D}$. To initialise $\boldsymbol{\Lambda}$, all problems are learned independently with the intermediate outputs (denoted as $\mathbf{Y}$ in Fig. 1) set to the expected responses, i.e. $\mathbf{R}$.

4.4. Analysis of the Algorithm

A number of observations on the dynamics of the proposed multi-target learning approach may be made. In the proposed structure learning technique, the interaction between the two groups of parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\Pi}$ is realised through the second-layer kernel matrix, i.e. through $\mathbf{J}$ (Step 6, Algorithm 1). In this respect, after updating the first-layer coefficient matrix $\boldsymbol{\Lambda}$, the intermediate outputs are generated via $\mathbf{Y} = \mathbf{K}\boldsymbol{\Lambda}$. The second-layer kernel matrix can then be produced by using the most recently updated intermediate outputs $\mathbf{Y}$ and θ. $\boldsymbol{\Pi}$ would then be computed using the recently updated $\mathbf{J}$. Any changes to $\boldsymbol{\Pi}$ would then affect the parameter $\boldsymbol{\Lambda}$ during the next iteration.

In the operational stage of the proposed approach, once a test observation $\mathbf{x}$ arrives, the intermediate responses (y_m's for $m = 1, \ldots, M$) for all tasks are generated through the initial layer. By considering the intermediate outputs as the elements of a vector $\mathbf{y} = [y_1, \ldots, y_M]^\top$, its similarity to the second layer training observations, i.e. to $\mathbf{y}_i$'s for $i = 1, \ldots, n$, is gauged using an RBF kernel, and finally, mixed together using the corresponding matrix $\boldsymbol{\Pi}$ to generate the final outputs.

4.5. Convergence

The loss function in the advocated approach in each step of the proposed alternating optimisation approach does not increase. This is true, since the optimisation w.r.t. $\boldsymbol{\Pi}$ is performed exactly, and hence, guaranteeing not to increase the loss function. By choosing suitable step sizes η_A and η_θ (e.g. via a line search), the optimisation of the loss function with respect to $\boldsymbol{\Lambda}$ and θ, is guaranteed not to increase it. As a consequence we have

$$\begin{aligned} \ldots \geqslant P(\boldsymbol{\Lambda}^t, \boldsymbol{\Pi}^t, \theta^t) \geqslant P(\boldsymbol{\Lambda}^{t+1}, \boldsymbol{\Pi}^t, \theta^t) \geqslant P(\boldsymbol{\Lambda}^{t+1}, \boldsymbol{\Pi}^t, \theta^{t+1}) \\ \geqslant P(\boldsymbol{\Lambda}^{t+1}, \boldsymbol{\Pi}^{t+1}, \theta^{t+1}) \geqslant \ldots \end{aligned} \tag{20}$$

As the data fidelity term is a Frobenius norm of a matrix, it is bounded from below by zero. Moreover, since the constraints

impose Tikhonov regularisation on task-specific coefficients (through matrix traces), they are also bounded by zero from below. Consequently, the loss function is bounded from below by zero. Therefore, the sequence generated by the proposed alternating optimisation is convergent in the limit by virtue of the theorem of monotone convergence.

5. Experimental Analysis

In this section, first, the datasets and the performance metric used in the experiments are introduced. Next, a comparison of the proposed Non-linear Output Structure Learning (denoted as 'NOSL') method to the state-of-the-art algorithms is presented and the convergence behaviour of the proposed method is analysed.

5.1. Datasets

The datasets used in the experiments are briefly introduced next.

5.1.1. Jura

The Jura [36] database incorporates measurements relating to a close gathering of 7 heavy metals including chromium, nickel, cadmium, zinc, cobalt, lead and copper that are measured in 359 different places in a region of Switzerland. The usage type of the land including Meadow, Forest, Tillage, Pasture, as well as the type of rock (such as Quaternary, Argovian, Portlandian, Sequanian, Kimmeridgian) are measured for each particular place. In a multi-target regression setting [37], one is interested in predicting the concentration of more expensive metals, which are considered as the primary variables based on the measurements of cheaper metals considered as input variables. In this work, copper, lead and cadmium are considered as targets and all the remaining metals in addition to the land usage type, type of rock and the locations of each place are employed as predictive inputs.

5.1.2. Slump

The concrete slump database [38] considers the prediction of three attributes of concrete, namely, flow, slump as well as compressive strength as a dependent vector variable of 7 concrete components including blast furnace slag, super plasticizer, cement, water, fly ash, fine aggregate and the coarse aggregate.

5.1.3. Andro

This database [39] considers the prediction of six future water quality values including the oxygen, pH, temperature, salinity, turbidity and conductivity in Thessaloniki, Greece. The target variable recordings are obtained from sensors placed under water whose sampling interval is nine seconds. These measurements are averaged to obtain a single record related to each variable on each specific day. The specific database that is commonly used is generated using a time-window of five days. In other words, the attributes relate to six water quality measurements for up to five previous days with a lead of five days. That is, the values of each variable for the six days ahead are predicted.

5.1.4. EDM

The Electrical Discharge Machining database [40] corresponds to a two-output regression task. The goal in this dataset is to make the machining performed faster via mimicking a human operator behaviours that supervises two output responses. Each output may take three different numeric values of -1, 0, or 1 and there exist sixteen continuous input features.

5.1.5. ENB

The Energy Building database [41] is concerned with the energy efficiency problem by predicting the heating and cooling load demands associated with buildings in terms of a function of 8 parameters such as roof area, overall height and glazing area, among others.

5.1.6. SCM20d

The Supply Chain Management database is gathered from the Trading Agent Competition in Supply Chain Management tournament. The data preprocessing and normalisation methods are detailed in [42]. The SCM20d dataset relates to the "Product Future" type of prediction. Each row in the dataset represents one observation day in the tournament where each game lasts for a total of 220 days and there are eighteen games in a tournament. The input features correspond to the observed prices for a single day in the tournament. Additionally, four time-delayed samples are incorporated for each observed component and product to facilitate anticipation of the ongoing trends. The SCM20d dataset for each product relates to the mean price over twenty coming days.

The statistics of the databases employed in this work are summarised in Table 1.

5.2. Performance Metric

In order to facilitate a benchmarking with state-of-the-art techniques, the performance of different approaches are gauged in terms of the Relative Root Mean Squared Error (RRMSE) which is computed as

$$RRMSE = \sqrt{\frac{\sum_{(x_j,y_j)\in D_{test}} (y_j - \hat{y}_j)^2}{\sum_{(x_j,y_j)\in D_{test}} (y_j - \bar{y})^2}} \tag{21}$$

where (x_j, y_j) denotes the j^{th} sample x_j with ground truth target y_j. The prediction of y_j is $\hat{y}_j$ and $\bar{y}$ corresponds to the average of the outputs over the training samples. The average RRMSE (aRRMSE) over all outputs within the test set is computed to provide a single performance metric for each algorithm. As the aRRMSE corresponds to an error estimate, a lower value for aRRMSE suggests a superior performance. The Tikhonov regularisation parameters γ_1 and γ_2 in the proposed approach are determined using cross validation on the training set from a grid of $10^{-5:1:2}$ by tuning one with the others fixed. The kernel function used for both layers in the proposed approach is a radial basis function.

5.3. Convergence Analysis

In this section, we examine the convergence characteristics of the proposed alternating optimisation approach. The convergence curves for two representative datasets, namely, the EBN and the Slump datasets are depicted in Fig. 2. As may be observed, the pro-

Table 1
Statistics of the datasets used.

Dataset	Samples	Input (d)	Target (T)
JURA	359	15	3
SLUMP	103	7	3
ANDRO	49	30	6
EDM	154	16	2
ENB	768	8	2
SCM20d	8966	61	16

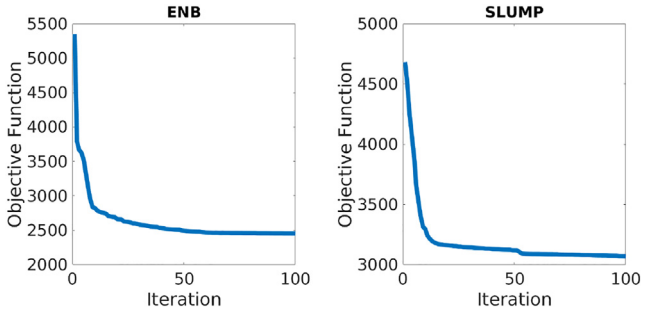

Fig. 2. Convergence curves corresponding to the proposed alternating optimisation approach on two sample datasets: ENB (left panel); SLUMP (right panel).

posed approach monotonically optimises the objective functions. During the initial iterations, the improvements in the cost functions are relatively notable, possibly due to large deviations of the initial parameters from their optimal values, which motivates taking larger steps in the negative direction of the gradient. As the optimisation method proceeds, the optimisation algorithm makes finer adjustments to the parameters, and hence, smaller changes in the objective function values are observed. The proposed optimisation approach converges within almost 100 iterations on both datasets. It should be noted that on the other datasets a similar behaviour has been observed.

5.4. Comparison to Other Techniques

The proposed technique is compared to other state-of-the-art multi-target regression methods in this section. These include multidimensional support vector regression (mSVR) [43], output kernel learning (OKL) [15], MROTS [20] and multi-task feature learning (MTFL) [44]. The multi-object random forests (MORF), single task learning (STL), the corrected multi-target stacking (MSTC), random linear target combinations (RLC) and ensemble of regressor chains (ERC) methods are also included to enable a comprehensive comparison due to their outstanding performance as observed in [9]. The multi-target regression drawing on low-rank learning (MMR) [12], which represents the state-of-the-art technique, is also included among the compared methods. The kernel ridge regression model (mKRR), which is formulated in the reproducing kernel Hilbert space and estimates each target independently, is included to serve as a baseline.

For a fair comparison, we follow the evaluation settings in [12] and use a 10-fold cross validation to benchmark against other algorithms. The results of the comparison to other methods are tabulated in Table 2. From this table, one may observe that on all databases examined the proposed method outperforms other methods. More specifically, compared to the best reported method, the proposed approach substantially improves the state-of-the-art on a number of datasets such as Andro and EDM. Specifically, while the previous best reported result on the Andro dataset was 52.7%, the aRRMSE of the proposed approach on this dataset is 35.3%. That is, more than thirty percent relative improvement in the aRRMSE. On the EDM dataset, the best reported result is 71.6% whereas the proposed method obtains an aRRMSE of 50.1% which again corresponds to more than a thirty percent improvement in the aRRMSE. Although the proposed formulation in this work which is based on Tikhonov regularisation demonstrates superior modelling capability, as future directions of investigation one may investigate other regularisation schemes to further boost the performance.

An average ranking of different algorithms using the Friedman's test is provided in Table 3. From the table it is evident that the proposed method (NOSL) ranks better than other alternatives. The second best performing method is MMR [12].

5.5. Computational Complexity

Updating the second-layer parameters $\mathbf{\Pi}$ requires a matrix inversion operation. A matrix inversion operation for an $n \times n$ matrix incurs a time complexity of $\mathcal{O}(n^3)$. Nevertheless, using the Sherman's March algorithm and the incremental Cholesky decomposition [45,46], the complexity of matrix inversion may be reduced to $\mathcal{O}(n^2)$. The gradient descent update for $\mathbf{\Lambda}$ and θ requires matrix multiplication operations with a time complexity of $\mathcal{O}(n^3)$. As a result, the complexity of the proposed approach scales cubically in the number of training observations.

However, the $O(n^3)$ complexity applies to a "serial" implementation of the matrix multiplication operation for the gradient descent update. One particularly advantageous feature of the proposed approach is that it may be implemented in a "parallel" fashion. In this context, the matrix multiplication operations may easily benefit from parallel processing units such as GPU's to obtain large speed-ups. Similarly, a parallel computation of the matrix inversion operation is viable to achieve remarkable gains in the computation time [47,48]. To examine this, the GPU and CPU computation times are measured for the matrix multiplication and for the matrix inversion operations for different numbers of observations on CPU and on GPU. The results are visualised in

Table 3
Mean ranks of different algorithms using the Friedman's test. ($p = 8.6502e-07$).

Algorithm	Ranking
MTSC	6.67
STL	5.91
RLC	5.25
ERC	4.58
mSVR	8.83
MORF	5.75
MTFL	10.00
MMR	2.16
OKL	6.83
MROTS	9.66
mKRR	11.33
NOSL	1.00

Table 2
Comparison of the proposed method with the state-of-the-art multi-target regression techniques on different datasets in terms of aRRMSE (%). (Best performances are indicated in bold)

	MTSC	STL	RLC	ERC	mSVR	MORF	MTFL	MMR	OKL	MROTS	mKRR	NOSL
JURA	59.1	58.9	59.6	59.0	61.1	59.7	60.8	58.2	59.9	62.5	63.3	**57.8**
SLUMP	69.5	68.8	69.0	68.9	71.1	69.4	68.1	58.7	69.9	77.8	78.9	**54.4**
ANDRO	57.9	60.2	57.0	56.7	62.7	51.0	80.3	52.7	55.3	63.5	63.9	**35.3**
EDM	74.0	74.2	73.5	74.1	73.7	73.4	85.1	71.6	74.1	81.2	83.3	**50.1**
ENB	12.1	11.7	12.0	11.4	22.0	12.1	31.6	11.1	13.8	25.7	26.3	**8.0**
SCM20d	47.5	47.5	44.3	39.4	49.3	48.2	64.3	38.9	44.3	45.6	49.8	**34.1**

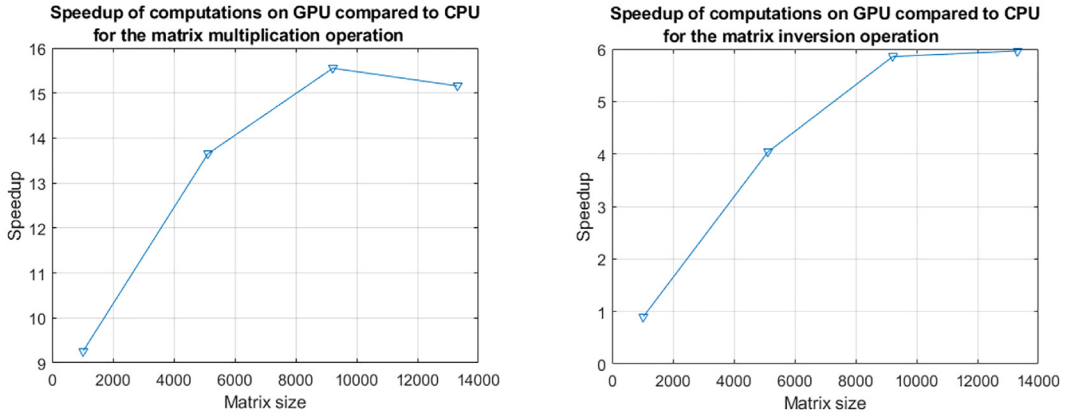

Fig. 3. Speedup gains obtained using a parallel processing of the proposed approach compared to a serial implementation..

Fig. 3 for a 64-bit machine with a 32 GB memory, 4 GHz CPU, using a GeForce GTX 1080Ti graphical processing unit operating in Matlab R2021a using Windows 10. From the figures, one may observe that a parallel processing on a GPU yields significant improvements in the running times corresponding to different steps of the proposed approach which enables the proposed method to be applied to larger sets of data. In particular, the speedup gain for the matrix multiplication operation reaches a 15× speedup for matrices larger than 8000-by-8000 while the relative speedup gain for the matrix inversion operation is also significant and reaches a 6× speedup. It should be noted that the speed-up gains observed correspond to a common GPU used on an ordinary PC. In problems of larger scales, an array of GPUs may be deployed to process a large amount of data which could deliver even larger speedup gains.

6. Conclusion

We considered the multi-target regression problem through learning a vector-valued composition function in a RKHS. In contrary to the existing approaches that try to capture inter-target correlations linearly, by virtue of a non-linear inter-target kernel, the proposed method facilitated non-linear structure learning among multiple outputs. For training, we presented an alternating minimisation approach with convergence guarantees. The experimental assessment of the proposed technique on standard multi-target regression datasets illustrated the benefits of the proposed approach as compared with existing approaches. The superior performance of the proposed technique can be attributed to the proposed non-linear learning of structural relationships between multiple targets which is able to captures real-world dependencies among multiple outputs considerably better.

Motivated by its widespread use as an effective regularisation mechanism, in this study, we employed a Tikhonov regularisation scheme. As future directions of investigation one may consider other regularisation mechanisms such as (r, p)-norms.

CRediT authorship contribution statement

Shervin Rahimzadeh Arashloo: Conceptualization, Methodology, Software. **Josef Kittler:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Kocev, S. Džeroski, M.D. White, G.R. Newell, P. Griffioen, Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition, Ecological Modelling 220 (8) (2009) 1159–1168, https://doi.org/10.1016/j.ecolmodel.2009.01.037.

[2] T. Xiong, Y. Bao, Z. Hu, Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting, Knowledge-Based Systems 55 (2014) 87–100, https://doi.org/10.1016/j.knosys.2013.10.012.

[3] D. Stojanova, M. Ceci, A. Appice, S. Džeroski, Network regression with predictive clustering trees, in: D. Gunopulos, T. Hofmann, D. Malerba, M. Vazirgiannis (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 333–348.

[4] M. Jeong, G.G. Lee, Multi-domain spoken language understanding with transfer learning, Speech Communication 51 (5) (2009) 412–424, https://doi.org/10.1016/j.specom.2009.01.001.

[5] Y. Yan, E. Ricci, R. Subramanian, G. Liu, O. Lanz, N. Sebe, A multi-task learning framework for head pose estimation under target motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (6) (2016) 1070–1083, https://doi.org/10.1109/TPAMI.2015.2477843.

[6] M. Emambakhsh, A. Bay, E. Vazquez, Convolutional recurrent predictor: Implicit representation for multi-target filtering and tracking, IEEE Transactions on Signal Processing 67 (17) (2019) 4545–4555, https://doi.org/10.1109/TSP.2019.2931170.

[7] Q. Liu, Q. Xu, V.W. Zheng, H. Xue, Z. Cao, Q. Yang, Multi-task learning for cross-platform siRNA efficacy prediction: an in-silico study, BMC Bioinformatics 11 (1) (2010) 181, https://doi.org/10.1186/1471-2105-11-181.

[8] M.M. Tatsuoka, Multivariate Analysis: Techniques for Educational and Psychological Research, 2nd Ed., Macmillan Publishing Co., Inc, USA, 1987.

[9] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: treating targets as inputs, Machine Learning 104 (1) (2016) 55–98, https://doi.org/10.1007/s10994-016-5546-z.

[10] A. Zaknich, Introduction to the modified probabilistic neural network for general signal processing applications, IEEE Transactions on Signal Processing 46 (7) (1998) 1980–1990, https://doi.org/10.1109/78.700969.

[11] J. Qi, J. Du, S.M. Siniscalchi, X. Ma, C.-H. Lee, Analyzing upper bounds on mean absolute errors for deep neural network-based vector-to-vector regression, IEEE Transactions on Signal Processing 68 (2020) 3411–3422, https://doi.org/10.1109/TSP.2020.2993164.

[12] X. Zhen, M. Yu, X. He, S. Li, Multi-target regression via robust low-rank learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2) (2018) 497–504, https://doi.org/10.1109/TPAMI.2017.2688363.

[13] C. Ciliberto, Y. Mroueh, T. Poggio, L. Rosasco, Convex learning of multiple tasks and their structure, in: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, pp. 1548–1557.

[14] A. Fawzi, M. Sinn, P. Frossard, Multitask additive models with shared transfer functions based on dictionary learning, IEEE Transactions on Signal Processing 65 (5) (2017) 1352–1365, https://doi.org/10.1109/TSP.2016.2634546.

[15] F. Dinuzzo, C.S. Ong, P. Gehler, G. Pillonetto, Learning output kernels with block coordinate descent, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), ICML '11, ACM, New York, NY, USA, 2011, pp. 49–56.

[16] X. Zhen, M. Yu, F. Zheng, I.B. Nachum, M. Bhaduri, D. Laidley, S. Li, Multitarget sparse latent regression, IEEE Transactions on Neural Networks and Learning Systems 29 (5) (2018) 1575–1586, https://doi.org/10.1109/TNNLS.2017.2651068.

[17] C. Brouard, M. Szafranski, F. d'Alché Buc, Supervised and semi-supervised structured output prediction with operator-valued kernels, Journal of Machine Learning Research 17 (176) (2016) 1–48. http://jmlr.org/papers/v17/15-602.html.

[18] R. Caruana, Multitask learning, Machine Learning 28 (1) (1997) 41–75, https://doi.org/10.1023/A:1007379606734.
[19] C.A. Micchelli, M. Pontil, On learning vector-valued functions, Neural Computation 17 (1) (2005) 177–204, https://doi.org/10.1162/0899766052530802.
[20] P. Rai, A. Kumar, H. Daume, Simultaneously leveraging output and task structures for multiple-output regression, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 25, Curran Associates Inc, 2012. https://proceedings.neurips.cc/paper/2012/file/4dcae38ee11d3a6606cc6cd636a3628b-Paper.pdf.
[21] A.J. Rothman, E. Levina, J. Zhu, Sparse multivariate regression with covariance estimation, Journal of Computational and Graphical Statistics 19 (4) (2010) 947–962, pMID: 24963268. doi:10.1198/jcgs.2010.09188.
[22] H. Liu, L. Wang, T. Zhao, Calibrated multivariate regression with application to neural semantic basis discovery, Journal of Machine Learning Research 16 (47) (2015) 1579–1606, http://jmlr.org/papers/v16/liu15b.html.
[23] L. Jacob, J.-P. Vert, F. Bach, Clustered multi-task learning: A convex formulation, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, Vol. 21, Curran Associates Inc, 2009. https://proceedings.neurips.cc/paper/2008/file/fccb3cdc9acc14a6e70a12f74560c026-Paper.pdf.
[24] Q. Zhou, Q. Zhao, Flexible clustered multi-task learning by learning representative tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2) (2016) 266–278, https://doi.org/10.1109/TPAMI.2015.2452911.
[25] T. Aho, B. Ženko, S. Džeroski, T. Elomaa, Multi-target regression with rule ensembles, Journal of Machine Learning Research 13 (78) (2012) 2367–2407, http://jmlr.org/papers/v13/aho12a.html.
[26] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of multi-objective decision trees, in: J.N. Kok, J. Koronacki, R.L. d. Mantaras, S. Matwin, D. Mladenič, A. Skowron (Eds.), Machine Learning: ECML 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 624–631.
[27] J.M. Moyano, O. Reyes, H.M. Fardoun, S. Ventura, Performing multi-target regression via gene expression programming-based ensemble models, Neurocomputing 432 (2021) 275–287, https://doi.org/10.1016/j.neucom.2020.12.060, https://www.sciencedirect.com/science/article/pii/S0925231220319603.
[28] X. Tian, Y. Li, T. Liu, X. Wang, D. Tao, Eigenfunction-based multitask learning in a reproducing kernel hilbert space, IEEE Transactions on Neural Networks and Learning Systems (2018) 1–13, https://doi.org/10.1109/TNNLS.2018.2873649.
[29] P. Li, S. Chen, Hierarchical gaussian processes model for multi-task learning, Pattern Recognition 74 (2018) 134–144, https://doi.org/10.1016/j.patcog.2017.09.021.
[30] M. Kordos, Álvar Arnaiz-González, C. García-Osorio, Evolutionary prototype selection for multi-output regression, Neurocomputing 358 (2019) 309–320. doi:https://doi.org/10.1016/j.neucom.2019.05.055. https://www.sciencedirect.com/science/article/pii/S0925231219307611.
[31] C.-H. Tu, C. Li, Multitarget prediction using an aim-object-based asymmetric neuro-fuzzy system: A novel approach, Neurocomputing 389 (2020) 155–169, https://doi.org/10.1016/j.neucom.2019.12.113. https://www.sciencedirect.com/science/article/pii/S0925231220300473.
[32] Z.-H. Feng, J. Kittler, X.-J. Wu, Mining hard augmented samples for robust facial landmark localization with cnns, IEEE Signal Processing Letters 26 (3) (2019) 450–454, https://doi.org/10.1109/LSP.2019.2895291.
[33] H. Borchani, G. Varando, C. Bielza, P. Larra naga, A survey on multi-output regression, WIREs Data Mining and Knowledge Discovery 5 (5) (2015) 216–233. doi: 10.1002/widm.1157.
[34] B. Bohn, M. Griebel, C. Rieger, A representer theorem for deep kernel learning, Journal of Machine Learning Research 20 (64) (2019) 1–32, http://jmlr.org/papers/v20/17-621.html.
[35] E. Alpaydin, Introduction to Machine Learning, 3rd Edition, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2014.
[36] A. Swan, Goovaerts, p. 1997. geostatistics for natural resources evaluation. applied geostatistics series. xiv 483 pp. new york, oxford: Oxford University Press. price £A3;46.95 (hard covers). isbn 0 19 511538 4, Geological Magazine 135 (6) (1998) 819-842. doi:10.1017/S0016756898631502.
[37] M.A. Álvarez, N.D. Lawrence, Computationally efficient convolved multiple output gaussian processes, Journal of Machine Learning Research 12 (41) (2011) 1459–1500, http://jmlr.org/papers/v12/alvarez11a.html.
[38] I.-C. Yeh, Modeling slump flow of concrete using second-order regressions and artificial neural networks, Cement and Concrete Composites 29 (6) (2007) 474–480, https://doi.org/10.1016/j.cemconcomp.2007.02.001.
[39] E.V. Hatzikos, G. Tsoumakas, G. Tzanis, N. Bassiliades, I. Vlahavas, An empirical study on sea water quality prediction, Knowledge-Based Systems 21 (6) (2008) 471–478, https://doi.org/10.1016/j.knosys.2008.03.005.
[40] A. Karaliccaron, I. Bratko, First order regression, Machine Learning 26 (1997) 147–176.
[41] A. Tsanas, A. Xifara, Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools, Energy and Buildings 49 (2012) 560–567, https://doi.org/10.1016/j.enbuild.2012.03.003.
[42] W. Groves, M. Gini, Improving prediction in tac scm by integrating multivariate and temporal aspects via pls regression, in: E. David, V. Robu, O. Shehory, S. Stein, A. Symeonidis (Eds.), Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 28–43.
[43] M. Sanchez-Fernandez, M. de Prado-Cumplido, J. Arenas-Garcia, F. Perez-Cruz, Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems, IEEE Transactions on Signal Processing 52 (8) (2004) 2298–2307, https://doi.org/10.1109/TSP.2004.831028.
[44] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, Vol. 19, MIT Press, 2007.
[45] G.W. Stewart, Matrix algorithms – Volume I: Basic decompositions, SIAM (2001).
[46] S.R. Arashloo, J. Kittler, Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features, IEEE Transactions on Information Forensics and Security 9 (12) (2014) 2100–2109, https://doi.org/10.1109/TIFS.2014.2359587.
[47] P. Benner, P. Ezzatti, E. Quintana-Ortí, A. Remón, Matrix inversion on cpu-gpu platforms with applications in control theory, Concurrency and Computation: Practice and Experience 25 (8) (2013) 1170–1182, https://doi.org/10.1002/cpe.2933.
[48] D. Yu, S. He, Y. Huang, G. Yu, L. Yang, A fast parallel matrix inversion algorithm based on heterogeneous multicore architectures, in: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2015, pp. 903–907. doi:10.1109/GlobalSIP.2015.7418328.

Shervin Rahimzadeh Arashloo received the Ph.D. degree from the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, U.K., in 2010. He is currently an Assistant Professor with the Department of Computer Engineering, Bilkent University, Ankara, Turkey, and a Visiting Research Fellow with CVSSP, University of Surrey. His research interests include pattern recognition, machine learning, and signal processing.

Josef Kittler (Life Member, IEEE) received the B.A., Ph.D., and D.Sc. degrees from the University of Cambridge, Cambridge, U.K., in 1971, 1974, and 1991, respectively. He is a Distinguished Professor of machine intelligence with the Center for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K. He conducts research in biometrics, video and image dataset retrieval, medical image analysis, and cognitive vision. He published the textbook Pattern Recognition: A Statistical Approach and over 700 scientific articles. His publications have been cited more than 68 000 times (Google Scholar).

Dr. Kittler is a Series Editor of Springer Lecture Notes on Computer Science. He currently serves on the Editorial Boards of Pattern Recognition Letters, Pattern Recognition and Artificial Intelligence, Pattern Analysis and Applications. He also served as a member of the Editorial Board of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 1982 to 1985. He served on the Governing Board of the International Association for Pattern Recognition (IAPR) as one of the two British representatives from 1982 to 2005, the President of the IAPR from 1994 to 1996. He is currently a member of the KS Fu Prize Committee of IAPR.