Proceedings of Machine Learning Research 77:407–422, 2017 ACML 2017

Magnitude-Preserving Ranking for Structured Outputs

Céline Brouard[1] CELINE.BROUARD@AALTO.FI
Eric Bach[1] ERIC.BACH@AALTO.FI
Sebastian Böcker[2] SEBASTIAN.BOECKER@UNI-JENA.DE
Juho Rousu[1] JUHO.ROUSU@AALTO.FI
[1] *Helsinki Institute for Information Technology, Department of Computer Science, Aalto University, Espoo, Finland*
[2] *Chair for Bioinformatics, Friedrich-Schiller University, Jena, Germany*

Editors: Yung-Kyun Noh and Min-Ling Zhang

Abstract

In this paper, we present a novel method for solving structured prediction problems, based on combining Input Output Kernel Regression (IOKR) with an extension of magnitude-preserving ranking to structured output spaces. In particular, we concentrate on the case where a set of candidate outputs has been given, and the associated pre-image problem calls for ranking the set of candidate outputs. Our method, called magnitude-preserving IOKR, both aims to produce a good approximation of the output feature vectors, and to preserve the magnitude differences of the output features in the candidate sets. For the case where the candidate set does not contain corresponding 'correct' inputs, we propose a method for approximating the inputs through application of IOKR in the reverse direction. We apply our method to two learning problems: cross-lingual document retrieval and metabolite identification. Experiments show that the proposed approach improves performance over IOKR, and in the latter application obtains the current state-of-the-art accuracy.

Keywords: Structured prediction, kernel methods

1. Introduction

Many real-world learning tasks require predicting outputs that correspond to complex structured objects or to multiple interdependent outputs. The basic approach of decomposing a structured prediction problem into simple problems that independently predict parts of structured objects is often inefficient and leads to deficient accuracy. Structured prediction approaches making use of the statistical dependencies between the output parts, have been shown to achieve an improved prediction performance in several applications, such as protein secondary structure prediction, hierarchical multilabel classification, natural language parsing, and metabolite identification.

In this paper, we focus on a kernel-based structured output prediction approach, called *output kernel regression* (Weston et al., 2003; Cortes et al., 2005; Geurts et al., 2006; Kadri et al., 2013; Brouard et al., 2016b). This approach is based on encoding the structure of the output data using a kernel function, and approximating the output feature map associated with this kernel through solving a regression problem. Kernel Dependency Estimation (Weston et al., 2003; Cortes et al., 2005; Kadri et al., 2013), Output Kernel Trees (Geurts et al., 2006) and Input Output Kernel Regression (IOKR) (Brouard et al., 2016b) are

methods belonging to this setting. These methods generally require solving a pre-image problem for extracting the models prediction, which in many cases is intractable (Giguère et al., 2015). Here, we focus on the frequent case, where no efficient pre-image algorithm is available, but a set of candidate outputs has been isolated, typically from the training set or through using human expert knowledge. In this case, the pre-image problem corresponds to a *ranking problem* in the candidate set.

In this work, we propose a new structured prediction method that augments the regression-based IOKR approach (Brouard et al., 2016b) so that the candidate set information can be used already in the learning phase, instead of the prediction phase only. This new method, called *magnitude-preserving IOKR*, looks to preserve the difference between training outputs and candidates in the output feature space. For this end, we extend to structured outputs the magnitude-preserving ranking method proposed by Cortes et al. (2007) for the learning problem of ranking. We show that the magnitude-preserving objective amounts to centering the inputs and outputs with respect to the candidate set means, thus encoding information of the candidate set to the input and output feature spaces. Moreover, we introduce a previously unstudied setting, that arises in structured output prediction tasks using the magnitude-preserving objective: the case where a set of candidate outputs can be defined or constructed, but the corresponding inputs are not known. For this case we propose an elegant approach to approximate the input feature vectors for the candidates. By doing that we can give a closed-form solution of the MP-IOKR objective function, even if the inputs of the candidates are unknown. In addition, when the input kernel is chosen as a linear combination of several kernels, we introduce two different approaches to approximate the corresponding input feature vectors.

2. Methods

We first describe the existing IOKR approach and then introduce the new magnitude-preserving IOKR framework. The notation used in this paper is described in Table 1.

2.1. Input Output Kernel Regression

Input Output Kernel Regression (IOKR) is an approach proposed by Brouard et al. (2011, 2016b) for learning mappings between a structured input set $\mathcal{X}$ and a structured output space $\mathcal{Y}$. In this approach, the internal structure of the output data is encoded using an output kernel function $k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. The problem of learning the mapping between $\mathcal{X}$ and $\mathcal{Y}$ is solved by first approximating the feature map ψ associated with the kernel k_y using a function h between the input set $\mathcal{X}$ and the output feature space $\mathcal{F}_y$. As the values of this function are vectors in $\mathcal{F}_y$, IOKR uses the RKHS theory devoted to vector-valued functions (Pedrick, 1957; Micchelli and Pontil, 2005). In this theory, the values of a kernel function $\mathcal{K}_x$ are operators from $\mathcal{F}_y$ to $\mathcal{F}_y$. Given a set S of ℓ training examples, the function h is searched in the RKHS associated with an operator-valued kernel $\mathcal{K}_x$ by solving the following optimization problem:

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{\ell} \|h(x_i) - \psi(y_i)\|^2_{\mathcal{F}_y} + \lambda\|h\|^2_{\mathcal{H}},\ \lambda > 0. \tag{1}$$

Table 1: Notation used in the paper

Symbol	Meaning
$\mathcal{X},\ \mathcal{Y}$	input, output sets
$k_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R},\ k_y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$	input, output scalar kernels
$\mathcal{F}_x,\ \mathcal{F}_y$	input, output feature spaces
$\phi : \mathcal{X} \to \mathcal{F}_x,\ \psi : \mathcal{Y} \to \mathcal{F}_y$	input, output feature maps
$\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \to \mathcal{B}(\mathcal{F}_y)$	input operator-valued kernel
$\mathcal{H}$	RKHS of $\mathcal{K}_x$
$K_X,\quad K_Y$	input, output Gram matrices
$S = \{1, \ldots, \ell\}$	set of training indices
C_i	candidate set of x_i
n_i	number of candidates in C_i
$n = \sum_{i=1}^{\ell} n_i$	total number of candidates
$C = \cup_{i=1}^{\ell} C_i$	union of the training candidate sets

When using the operator-valued kernel $\mathcal{K}_x(x, x') = k_x(x, x')I$, where $k_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a scalar-valued kernel and I is an identity operator, the solution of Equation (1) can be written:

$$h(x) = \Psi_S(\lambda I_\ell + K_{X_S})^{-1} k^x_{X_S}, \tag{2}$$

where $\Psi_S \in \mathbb{R}^{|\mathcal{F}_y| \times \ell}$ is a matrix defined by $\Psi_S = [\psi(y_1), \ldots, \psi(y_\ell)]$. K_{X_S} denotes the Gram matrix of the kernel k_x on the training set and $k^x_{X_S}$ is the vector defined by $k^x_{X_S} = [k_x(x_1, x), \ldots, k_x(x_\ell, x)]^T$.

The predicted feature vector $h(x)$ is then mapped back to the output space $\mathcal{Y}$ by solving a pre-image problem. The pre-image problem is solved by determining the structured output $y \in \mathcal{Y}$ for which the distance between $\psi(y)$ and $h(x)$ in the output feature space is minimal:

$$f(x) = \operatorname*{argmin}_{y \in \mathcal{Y}} \|h(x) - \psi(y)\|^2_{\mathcal{F}_y}. \tag{3}$$

When replacing $h(x)$ by the solution given in Equation (2) and using the kernel trick in the output space, the expression of the pre-image problem becomes:

$$f(x) = \operatorname*{argmin}_{y \in \mathcal{Y}} k_y(y, y) - 2(k^y_{Y_S})^T (\lambda I_\ell + K_{X_S})^{-1} k^x_{X_S},$$

where $k^y_{Y_S} = [k_y(y_1, y), \ldots, k_y(y_\ell, y)]^T \in \mathbb{R}^{\ell \times 1}$.

Efficient pre-image algorithms exist for some particular structures, for example in hierarchical classification. In this work, we focus on the often experienced case where no efficient pre-image algorithm exists. Instead we assume that for each example x_i a set containing potential candidate outputs $\{y_j\}_{j \in C_i}$ can be defined and the pre-image algorithm is an exhaustive search among this candidate set. Several previous works have taken the outputs occurring in the training set as candidate set. Such a set could also be extracted by using human expert knowledge or by doing a local search around a seed output.

In this case, the pre-image problem reduces to a ranking problem of these candidates, where the candidates are sorted according to their distance to the predicted output feature vector $h(x)$. A good prediction is obtained in the case where a small rank is assigned to the correct candidate.

2.2. Magnitude-preserving IOKR

Regression based methods for structured outputs, such as IOKR, learn a function that approximates the output feature map, however, the ranking problem encountered in the pre-image step is not taken into account in the learning phase. Thus, the information in the candidate set is not taken advantage in model training. Here we introduce a new method, called *magnitude-preserving input-output kernel regression* (MP-IOKR), that incorporates the information of the candidate ranking when approximating the output feature vectors, through combining the idea of the magnitude-preserving ranking with IOKR.

The MP-IOKR approach learns a mapping between two structured sets by approximating a feature map associated with an output kernel and by mapping this approximation back to the output space by solving a pre-image problem. Compared to IOKR, in MP-IOKR the first step is augmented to take into account the candidate set used when solving the pre-image step for each input. Given a sample of ℓ training examples $\{(x_i, \psi(y_i)) \in \mathcal{X} \times \mathcal{F}_y\}_{i=1}^{\ell}$, we consider the following objective function to be minimized:

$$\mathcal{J}(h) = \sum_{i=1}^{\ell} \frac{1}{n_i} \sum_{j \in C_i} \|(h(x_i) - h(x_j)) - (\psi(y_i) - \psi(y_j))\|^2_{\mathcal{F}_y} + \lambda \|h\|^2_{\mathcal{H}}, \tag{4}$$

where $\lambda > 0$ is a regularization parameter. The set C_i contains the indices of the candidates for the training example x_i and $n_i = |C_i|$ corresponds to the number of candidates in this set. The objective function penalizes discrepancy between the pairwise differences of predictions $h(x_i) - h(x_j)$ and the pairwise differences of the ground truth $\psi(y_i) - \psi(y_j)$. This extends the magnitude-preserving ranking approach proposed by Cortes et al. (2007) for learning ranking. A similar approach, called RankRLS, was proposed at the same time by Pahikkala et al. (2007). However, in our case the considered targets are vectors in the output feature space rather than scalars, e.g. ratings, and the magnitudes are taken between a training example and each of its candidates.

The solution h of this optimization problem is searched in the RKHS $\mathcal{H}$ associated with an operator-valued kernel $\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \to \mathcal{B}(\mathcal{F}_y)$. We state a representer theorem for the MP-IOKR optimization problem.

Theorem 1 *The solution of the optimization problem* (4) *admits a representation of the form:*

$$\forall x \in \mathcal{X},\ h(x) = \sum_{i \in S \cup C} \mathcal{K}_x(x, x_i)\mathbf{c}_i,\ \mathbf{c}_i \in \mathcal{F}_y.$$

The proof of this theorem is given in the supplementary materials.

In the following we consider the operator-valued kernel

$$\mathcal{K}_x(x, x') = k_x(x, x')I, \tag{5}$$

where $k_x : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a scalar-valued kernel and I the identity operator. The scalar kernel is associated with a feature space $\mathcal{F}_x$ and a feature map $\phi : \mathcal{X} \to \mathcal{F}_x$. Using this kernel allows us to work in the general setting, where the dimension of the output feature space can be infinite and output feature vectors are not explicitly known. Using this kernel, the expression of the function h can be rewritten as $h(x) = W\phi(x)$, where W is a linear operator from $\mathcal{F}_x$ to $\mathcal{F}_y$.

We show that the optimization problem in Equation (4) can be casted back to an IOKR optimization problem on the training and candidate examples with modified input and output feature vectors. This modification consists in centering the input/output feature vectors around the input/output feature center of one of the candidate sets, this set varying depending on the examples. In the following, we note $S = \{1, \ldots, \ell\}$ the set containing the indices of the training examples and $C = \cup_{i=1}^{\ell} C_i$, the union of the training candidate sets.

Theorem 2 *When using the operator-valued kernel defined in Equation (5), the optimization problem (4) can be rewritten under the following form:*

$$\min_W \sum_{j \in S \cup C} \|W\phi'(x_j) - \psi'(y_j)\|^2_{\mathcal{F}_y} + \lambda \|W\|^2_F. \tag{6}$$

The modified input feature vectors are defined as:

$$\phi'(x_j) = \begin{cases} \phi(x_j) - \overline{\phi}_{C_j} & \text{if } j \in S \\ \frac{1}{\sqrt{n_i}} \left(\phi(x_j) - \overline{\phi}_{C_i}\right) & \text{if } j \in C_i \end{cases}, \tag{7}$$

where $\overline{\phi}_{C_i} = \frac{1}{n_i} \sum_{j \in C_i} \phi(x_j)$. *The modified output feature vectors are defined similarly.*

In the following, we derive the solution of this optimization problem. Let Ψ_S and Φ_S be the matrices containing the training input/output feature vectors: $\Psi_S = [\psi(y_1), \ldots, \psi(y_\ell))]$ and $\Phi_S = [\phi(x_1), \ldots, \phi(x_\ell))]$. We note similarly Φ_{C_i} and Ψ_{C_i} the matrices containing respectively the input and output feature vectors for the candidates belonging to the set C_i. Finally we define Ψ_C the matrix defined as $\Psi_C = [\Psi_{C_1}, \ldots, \Psi_{C_\ell}]$ and $\bar{\Psi}_C = [\bar{\psi}_{C_1}, \ldots, \bar{\psi}_{C_\ell}]$. We define similarly the matrices Φ_C and $\bar{\Phi}_C$ for the input feature vectors.

Proposition 3 *Using the solution of the IOKR optimization problem in Equation (2), we obtain the following expression for the function minimizing the objective function in Equation (6):*

$$h(x) = \Psi' \left(\lambda I_{\ell+n} + \Phi'^T \Phi'\right)^{-1} \Phi'^T \phi(x),$$

where Ψ' *is the matrix defined as* $\Psi' = [\Psi'_S, \Psi'_C] = \left[\Psi_S - \bar{\Psi}_C, (\Psi_C - \bar{\Psi}_C V^T) D_n\right]$ *and* $\Phi' = [\Phi'_S, \Phi'_C] = \left[\Phi_S - \bar{\Phi}_C, (\Phi_C - \bar{\Phi}_C V^T) D_n\right]$. V *is a matrix of size* $n \times \ell$, *where* $n = \sum_{i=1}^{\ell} n_i$, *defined such that:* $V_{ij} = 1$ *if* $i \in C_j$ *and* $V_{ij} = 0$ *otherwise.* $D_n \in \mathbb{R}^{n \times n}$ *is a diagonal matrix defined such that:* $[D_n]_{ii} = \frac{1}{\sqrt{n_j}}$ *if* $i \in C_j$.

The computation of this solution requires inverting a matrix of size $(\ell + n) \times (\ell + n)$.

The terms $\Phi'^T \Phi'$ and $\Phi'^T \phi(x)$ can be expressed in term of kernel values: $\Phi'^T \phi(x) = \begin{bmatrix} k^x_{X_S} - V^T D_n^2 k^x_{X_C} \\ D_n(I_n - V V^T D_n^2) k^x_{X_C} \end{bmatrix}$ and $\Phi'^T \Phi' = \begin{bmatrix} K'_{X_S} & K'_{X_{S,C}} \\ (K'_{X_{C,S}})^T & K'_{X_C} \end{bmatrix}$ where

- $K'_{X_S} = K_{X_S} - K_{X_{S,C}} D_n^2 V - (K_{X_{S,C}} D_n^2 V)^T + V^T D_n^2 K_{X_C} D_n^2 V$,
- $K'_{X_{S,C}} = (K_{X_{S,C}} - V^T D_n^2 K_{X_C})(I_n - D_n^2 V V^T) D_n$,
- $K'_{X_C} = D_n (I_n - V V^T D_n^2) K_{X_C} (I_n - D_n^2 V V^T) D_n$.

2.3. Approximating the Inputs of the Candidates

The computation of the input kernel matrices $\Phi'^T \Phi'$ and $\Phi'^T \phi(x)$ requires knowing the input features corresponding to the candidate outputs. However, this information might not be available in all applications, for example if the candidate sets are large and the generation of the inputs corresponding to the outputs is expensive. In the experiments section, we consider the metabolite identification problem, in which the inputs are tandem mass spectra available for a few thousand of metabolites, while millions of molecular candidates are available. Generating a reference tandem mass spectrum for a new molecule requires a heavy experimental protocol which is prohibitively expensive for the sole purpose of generating training data for machine learning algorithms.

In the following, we address this problem by approximating the candidate input feature vectors from their outputs. We first address the case of inputs being represented by a single input kernel and then the case of a linear combination of input kernels.

2.3.1. Single Input Kernel Case

We propose to approximate the input feature vector $\phi(x_i)$ of a candidate from its output y_i using a function $g : \mathcal{Y} \to \mathcal{F}_x$. In this case, the expression of ϕ' defined in Equation (7) becomes:

$$\phi'(x_j) = \begin{cases} \phi(x_j) - \overline{g}_{C_j} & \text{if } j \in S \\ \frac{1}{\sqrt{n_i}} \left(g(y_j) - \overline{g}_{C_i} \right) & \text{if } j \in C_i \end{cases}. \tag{8}$$

The definition of ψ' remains the same as in the previous subsection.

As shown in Section 2.1, the first step of IOKR consists in learning an output feature map. The function g can thus be learned by minimizing a similar objective:

$$\min_{g \in \mathcal{G}} \sum_{i=1}^{\ell} \|g(y_i) - \phi(x_i)\|_{\mathcal{F}_y}^2 + \gamma \|g\|_{\mathcal{G}}^2, \ \gamma > 0. \tag{9}$$

According to the Representer theorem for vector-valued functions, the function g admits the following expansion: $\forall y \in \mathcal{Y}$, $g(y) = \sum_{i=1}^{\ell} \mathcal{K}_y(y, y_i) \mathbf{b}_i$, where $\mathcal{K}_y : \mathcal{Y} \times \mathcal{Y} \to \mathcal{B}(\mathcal{F}_x)$ is an operator-valued kernel. Using the IOKR approach with the operator-valued kernel $\mathcal{K}_y(y, y') = k_y(y, y') I$, the expression obtained for the function g is:

$$g(y) = \Phi_S (\gamma I_\ell + K_{Y_S})^{-1} k_{Y_S}^y. \tag{10}$$

In the following we note $M = (\gamma I_\ell + K_{Y_S})^{-1}$.

Proposition 4 *The solution of the optimization problem (6) when using the expression of ϕ' in Equation (8) is given by:*

$$h(x) = \Psi' A^T \left(\lambda I_\ell + K_{X_S} A A^T \right)^{-1} k_{X_S}^x, \tag{11}$$

where $A = \left[I_\ell - M\Psi_S^T\bar{\Psi}_C, M\Psi_S^T\Psi'_C\right]$.

The proof is given in the supplementary materials. The computation of this solution only requires inverting a matrix of size $\ell \times \ell$, while it is $(\ell+n)\times(\ell+n)$ in the case where the candidate input feature vectors can be obtained. Regarding the computational complexity, the dominant term is $\mathcal{O}(\ell^2(\ell+n))$.

2.3.2. Multiple Input Kernels Case

We now consider the case where data from different sources are available. We note $\{k_x^i\}_{i=1}^K$ the K kernels associated with these different input sources and we consider a linear combination of them as input kernel: $k_x(x,x') = \sum_{i=1}^K \mu_i k_x^i(x,x')$, $\mu_i \geq 0$. For the kernel weights, in this paper we use uniform weighting $\mu_i = 1/K$, for $i = 1,\ldots,K$, but we note that in general some multiple kernel learning algorithm could be used to learn the weights.

For $j \in \{1,\ldots K\}$, let $\phi^j(x_i)$ be a candidate input feature vector associated with the kernel k_x^j. Then $\phi(x_i) = [\sqrt{\mu_1}\phi^1(x_i),\ldots,\sqrt{\mu_K}\phi^K(x_i)]^T$ is an input feature vector associated with the combined kernel k_x. If we directly apply the candidate input feature approximation approach described in 2.3.1, then according to Equation (10), an approximation of this feature vector will write as:

$$g(y_i) = \Phi_S(\gamma I_\ell + K_{Y_S})^{-1}k_{Y_S}^{y_i},$$

where $\Phi_S = [\sqrt{\mu_1}\Phi_S^1,\ldots,\sqrt{\mu_K}\Phi_S^K]^T$.

We propose an alternative approach, in which the input feature vectors of the candidates are approximated separately for each base kernel:

$$g(y_i) = \begin{bmatrix} \sqrt{\mu_1}\Phi_S^1(\gamma_1 I_\ell + K_{Y_S})^{-1}k_{Y_S}^{y_i} \\ \vdots \\ \sqrt{\mu_K}\Phi_S^K(\gamma_K I_\ell + K_{Y_S})^{-1}k_{Y_S}^{y_i} \end{bmatrix}. \tag{12}$$

The matrix containing the approximated input feature vectors of all the candidates writes as:

$$\tilde{\Phi}_C = D_{\Phi_S}\mathbf{M}K_{Y_S,Y_C},$$

where D_{Φ_S} is a $K\times K$ block diagonal matrix such that $[D_{\Phi_S}]_{jj} = \sqrt{\mu_j}\Phi_S^j$ and $\mathbf{M}$ is a $K\times 1$ block matrix where $M_j = (\gamma_j I_\ell + K_{Y_S})^{-1}$. K_{Y_S,Y_C} is the output Gram matrix between the training and the candidate sets.

Proposition 5 *The solution of the optimization problem (6) when using the function g defined in Equation (12) writes as:*

$$h(x) = \Psi' A^T\left(\lambda I_{\ell K} + D_{K_{X_S}}AA^T\right)^{-1}k_{X_S}^x, \tag{13}$$

where $A = \left[\mathbf{I} - \mathbf{M}\Psi_S^T\bar{\Psi}_C, \mathbf{M}\Psi_S^T\Psi'_C\right]$, $D_{K_{X_S}} = diag(\mu_1 K_{X_S}^1,\ldots,\mu_k K_{X_S}^K)$ *and* $\mathbf{I} = [I_\ell,\ldots,I_\ell]^T$ *is a matrix of size $\ell K \times \ell$. $k_{X_S}^x$ now denotes a vector of length ℓK defined as:* $k_{X_S}^x = [(k^{1x}_{X_S})^T,\ldots,(k^{Kx}_{X_S})^T]^T$.

The derivation of this solution is provided in the supplementary materials. We notice that the size of the matrix to invert is $\ell K \times \ell K$ and depends on the number of training data and of the number of base kernels. The dominant term for the computation of this solution is $\mathcal{O}((\ell K)^2(\ell K + \ell + n))$. This approach can be very heavy in the case where the number of kernels K is large. In practice, the regularization parameters γ_j are often selected among a finite set of parameters. If the selected parameter is the same for a set of kernels, then the computational complexity can be reduced by grouping these kernels. This is done by considering only one matrix M_j for a selected regularization parameter γ_j in $\mathbf{M}$ and by using the linear combination of the kernels associated with γ_j as corresponding input kernel in $D_{K_{X_S}}$.

3. Experiments

We evaluated our developed methods in two representative applications, that also demonstrate the generality of the methods for different structured prediction problems. The first application, cross-lingual document retrieval, deals with the case where the inputs corresponding to the candidate outputs are available. The second application, metabolite identification, concerns the case where the inputs are not available and need to be approximated.

3.1. Cross-lingual Document Retrieval

We performed experiments on a cross-lingual document retrieval task. Given a document written in one language, the goal of this task is to retrieve the translation of this document within a document corpus written in a different language. The inputs are therefore the documents written in the source language and the outputs are the translation of these documents in the target language. In this case, the candidate set for our method is the corpus of documents written in the target language. During the learning phase of MP-IOKR, we used the outputs occurring in the training set as candidate set. The candidate inputs are therefore known in this setting and we used the approach described in Section 2.2.

We used as dataset a subset of the JRC-Acquis multilingual parallel corpus (Steinberger et al., 2006). This corpus contains legal documents from the European Union (EU) which have been translated and aligned in the different official languages of the EU. We considered a subset of 10,000 aligned documents in French, English, Spanish, German and Dutch languages. We processed the documents of the corpus with the quanteda R package, including tokenization, removal of punctuation and stop worlds and stemming. We then computed a document feature representation using a term frequency inverse document frequency model (TF-IDF). We built linear kernels between the TF-IDF representations of the documents for the input and output kernels.

We randomly selected 5000 documents for the training set and 5000 for the test set. The performances were averaged over ten random partitions of the documents in training and test sets. The regularization parameter λ was selected using a 5-fold cross-validation experiment on the training set among the set $[10^{-7}, 10^{-6}, \ldots, 10^{4}, 10^{5}]$. For evaluating the performance, we computed the top-k accuracy, which corresponds to the percentage of test examples for which the correct answer is found among the k top ranked candidates. We performed a Welch's t-test for the different input/output language pairs. The top-k accuracies obtained with IOKR and MP-IOKR, as well as the corresponding p-values

Table 2: Top-k accuracies obtained for each pair of source (S) and target (T) languages in the cross-lingual document retrieval task using IOKR and MP-IOKR. Colored cells indicate significant improvement according to the Welch's t-test. Shades of blue are used for representing p-values lower than the chosen significance level ($p = 0.05$). The following shortcuts are used for the languages: fr: French, en: English, es: Spanish, de: German, nl: Dutch.

Language		**Top-1**		**Top-5**		**Top-10**	
S	T	IOKR	MP-IOKR	IOKR	MP-IOKR	IOKR	MP-IOKR
fr	en	55.1 ± 0.6	$\mathbf{60.7 \pm 0.6}$	84.5 ± 0.6	$\mathbf{89.7 \pm 0.4}$	90.7 ± 0.4	$\mathbf{94.2 \pm 0.3}$
	es	58.5 ± 2.2	$\mathbf{63.0 \pm 0.7}$	86.0 ± 1.9	$\mathbf{90.0 \pm 0.4}$	91.4 ± 1.5	$\mathbf{94.4 \pm 0.3}$
	de	32.9 ± 0.5	$\mathbf{35.3 \pm 0.6}$	74.4 ± 0.7	$\mathbf{81.1 \pm 0.6}$	84.4 ± 0.5	$\mathbf{89.5 \pm 0.3}$
	nl	36.3 ± 0.5	$\mathbf{38.9 \pm 0.6}$	76.4 ± 0.5	$\mathbf{83.0 \pm 0.3}$	85.6 ± 0.4	$\mathbf{90.6 \pm 0.3}$
en	fr	52.3 ± 0.7	$\mathbf{57.2 \pm 0.7}$	84.5 ± 0.5	$\mathbf{89.4 \pm 0.3}$	90.6 ± 0.4	$\mathbf{94.0 \pm 0.3}$
	es	54.6 ± 0.7	$\mathbf{59.5 \pm 0.7}$	84.3 ± 0.6	$\mathbf{89.2 \pm 0.4}$	90.3 ± 0.5	$\mathbf{93.9 \pm 0.3}$
	de	33.4 ± 0.5	$\mathbf{34.9 \pm 0.5}$	74.7 ± 0.6	$\mathbf{81.1 \pm 0.5}$	84.5 ± 0.5	$\mathbf{89.4 \pm 0.4}$
	nl	36.5 ± 0.6	$\mathbf{38.4 \pm 0.6}$	76.4 ± 0.6	$\mathbf{82.8 \pm 0.4}$	85.6 ± 0.4	$\mathbf{90.3 \pm 0.4}$
es	fr	55.9 ± 0.5	$\mathbf{60.8 \pm 0.5}$	85.4 ± 0.5	$\mathbf{90.0 \pm 0.3}$	91.1 ± 0.4	$\mathbf{94.5 \pm 0.3}$
	en	55.6 ± 0.7	$\mathbf{60.9 \pm 0.6}$	84.5 ± 0.6	$\mathbf{89.5 \pm 0.3}$	90.5 ± 0.3	$\mathbf{94.0 \pm 0.3}$
	de	33.5 ± 0.6	$\mathbf{35.1 \pm 0.5}$	74.2 ± 0.6	$\mathbf{80.6 \pm 0.5}$	84.3 ± 0.4	$\mathbf{89.1 \pm 0.4}$
	nl	36.7 ± 0.6	$\mathbf{38.5 \pm 0.5}$	76.3 ± 0.6	$\mathbf{82.7 \pm 0.3}$	85.7 ± 0.4	$\mathbf{90.3 \pm 0.2}$
de	fr	43.7 ± 0.6	$\mathbf{44.1 \pm 0.6}$	83.9 ± 0.5	$\mathbf{84.4 \pm 0.5}$	90.8 ± 0.4	$\mathbf{91.1 \pm 0.4}$
	en	47.6 ± 0.7	$\mathbf{48.1 \pm 0.7}$	84.9 ± 0.5	$\mathbf{85.5 \pm 0.5}$	91.3 ± 0.4	$\mathbf{91.6 \pm 0.4}$
	es	45.9 ± 0.7	$\mathbf{46.4 \pm 0.7}$	84.0 ± 0.5	$\mathbf{84.4 \pm 0.6}$	90.5 ± 0.5	$\mathbf{90.9 \pm 0.5}$
	nl	36.5 ± 1.3	$\mathbf{37.4 \pm 0.6}$	81.0 ± 2.2	$\mathbf{82.7 \pm 0.5}$	89.2 ± 1.5	$\mathbf{90.3 \pm 0.4}$
nl	fr	47.9 ± 0.5	$\mathbf{48.3 \pm 0.4}$	85.3 ± 0.4	$\mathbf{85.8 \pm 0.4}$	91.6 ± 0.3	$\mathbf{91.9 \pm 0.2}$
	en	51.1 ± 0.6	$\mathbf{51.8 \pm 0.6}$	85.9 ± 0.4	$\mathbf{86.4 \pm 0.3}$	91.8 ± 0.4	$\mathbf{92.2 \pm 0.4}$
	es	50.0 ± 0.7	$\mathbf{50.6 \pm 0.7}$	85.3 ± 0.5	$\mathbf{85.8 \pm 0.5}$	91.6 ± 0.5	$\mathbf{91.9 \pm 0.5}$
	de	33.7 ± 0.5	$\mathbf{36.9 \pm 0.5}$	76.3 ± 0.6	$\mathbf{82.7 \pm 0.5}$	85.8 ± 0.5	$\mathbf{90.4 \pm 0.3}$

$p < 0.0001$	$0.0001 \leq p < 0.001$	$0.001 \leq p < 0.01$	$0.01 \leq p < 0.05$	$p \geq 0.05$

are shown in Table 2. We first observe that MP-IOKR consistently obtains better top-k accuracies compared to IOKR for the different language pairs and values of k. We also observe that this improvement is significant for 15 to 19 language pairs out of 20 depending on the value of k. IOKR and MP-IOKR both present better performance when the chosen target language is French, English and Spanish compared to German and Dutch.We also note that the relative improvement observed for MP-IOKR compared to IOKR is larger for German and Dutch, this means for the most difficult target languages in this application.

3.2. Metabolite Identification

Our second application is the metabolite identification problem, which is an important task in metabolomics. Metabolites are small molecules involved in the biological processes of organisms. Given a biological sample, e.g. cells, blood or other biofluids, the task is to determine the molecular structures of the unknown metabolites contained in the sample. Mass spectrometry (MS) is a popular method to extract features from biological samples due

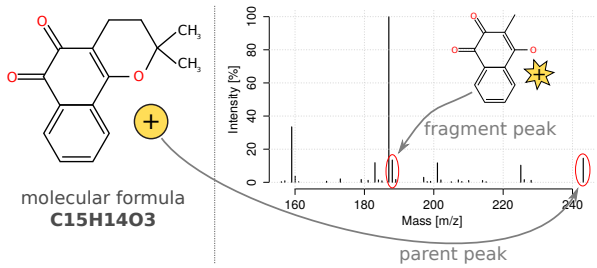

Figure 1: Molecular structure (left) and tandem mass spectrum (right) of the metabolite `beta-Lapachone`. The x-axis shows the mass-to-charge ratio of the measured fragments and the y-axis the (relative) abundance of the fragments.

to its sensitivity and applicability for a wide range of metabolites (Patti et al., 2012). A mass spectrometer measures the abundance of charged molecules or molecular fragments with a certain mass-to-charge ratio. Using the MS technique, a tandem mass (MS/MS) spectrum can be extracted for a molecule in the sample by fragmenting it and measuring the obtained fragments with the mass spectrometer. The resulting MS/MS spectrum contains a set of peaks associated with the fragments of the measured molecule. The mass of a fragment determines its peak's position and its abundance the peak's height. An example MS/MS spectrum is shown in Figure 1.

Dataset. We used a dataset containing 4138 annotated MS/MS spectra of metabolites extracted from the GNPS (Global Natural Products Social) public spectral library (Wang et al., 2016). We considered 24 different input kernels that were previously used by Brouard et al. (2016a) to solve the metabolite identification task with the IOKR approach. These kernels are defined based on the MS/MS spectra and on the fragmentation trees (Böcker and Rasche, 2008), which can be computed from the MS/MS spectra. The description of these kernels can be found in Brouard et al. (2016a). We combined the different input kernels uniformly. On the output side, we measured the similarity between the molecules using a linear kernel between their molecular fingerprints. These fingerprints are binary vectors where each bit indicates the presence or absence of a certain molecular property. We considered fingerprints built from a set of 2765 molecular properties, which are described in Brouard et al. (2016a).

For each metabolite in the dataset, the corresponding candidate set contained all the molecular structures in the molecular database PubChem (Kim et al., 2016) having the same molecular formula as the considered metabolite (see Figure 1 for an example of a molecular formula). As in Brouard et al. (2016a), the molecular formula of the test examples were supposed to be known.

Experiment setup. We compared MP-IOKR to IOKR (Brouard et al., 2016a) and to two competing methods: CSI:FingerID (Dührkop et al., 2015; Shen et al., 2014) and CFM-ID (Allen et al., 2015). CSI:FingerID trains separate SVMs for each binary molecular property. For this method we used the same kernels, combined using the ALIGNF multiple kernel learning method (Cortes et al., 2012), and the same molecular fingerprints and

Table 3: Top-k accuracy for the metabolite identification task using different methods. For MP-IOKR the left value results from using all the candidates and the right one from using 1% randomly chosen candidates. *(j)* indicates the joint input feature approximation and *(s)* the separate one. The standard deviation of the top-k accuracy in the random selection case was 0.1 for the three reported values.

Method	**Top-1**	**Top-10**	**Top-20**
CFM-ID	14.8	46.6	55.9
CSI:FingerID	29.7	64.3	71.7
IOKR	30.7	66.3	73.9
MP-IOKR (j)	30.7/30.8	67.0/67.2	74.6/74.6
MP-IOKR (s)	**31.2**/**31.2**	**67.9**/67.8	**75.3**/**75.3**

candidate sets as for MP-IOKR. CFM-ID is a probabilistic approach simulating MS/MS spectra. Given an input MS/MS spectrum the spectra of its candidates are simulated using their molecular structure. The input is then compared with all simulated spectra and the predicted molecular structure is chosen as the one corresponding to the most similar spectrum. We trained a single-energy model using the latest version of the CFM-ID software[1]. The training parameters were chosen as suggested in Allen et al. (2015). We excluded 270 spectra from the training as their corresponding molecular structures can not be processed using the CFM-ID software.

We performed the evaluation using a 10-fold cross-validation (CV) experiment. In the evaluation of the predictive performance we considered a subset of 3868 examples, which could be processed by all methods we compared MP-IOKR with.

For MP-IOKR we first selected the γ_i parameters of the input feature approximation and subsequently the λ parameter on the same training set. The γ_i and λ parameters where selected using cross-validation on the training set. As in this application we did not have the inputs (MS/MS spectra) for each molecule in the candidate sets, we used the approximation method described in Section 2.3. When we used the approach to separately approximate the input features, we observed that only 5 different γ_i were selected as regularization parameters. We could therefore group the input kernels into 5 groups as described in Section 2.3.2. During the training phase of MP-IOKR we used two alternative strategies for using the candidate sets: either all candidates or a random subset of 1% of the candidates corresponding to the training examples were used. The first set contained around 5.8 million (of which 2.5 million have a unique molecular structure) and the latter around 58000 molecular structures. The random selection of the candidates and MP-IOKR training was repeated 20 times, and averaged results are reported.

Results. In Table 3 we summarize the results for the metabolite identification task and in Figure 2 we show the performance difference of MP-IOKR and CSI:FingerID compared with IOKR on the range of top-1 to top-100. We omitted the curve for MP-IOKR using a random subset of candidates in the figure as it follows the one using all the candidates. We

1. https://sourceforge.net/projects/cfm-id/

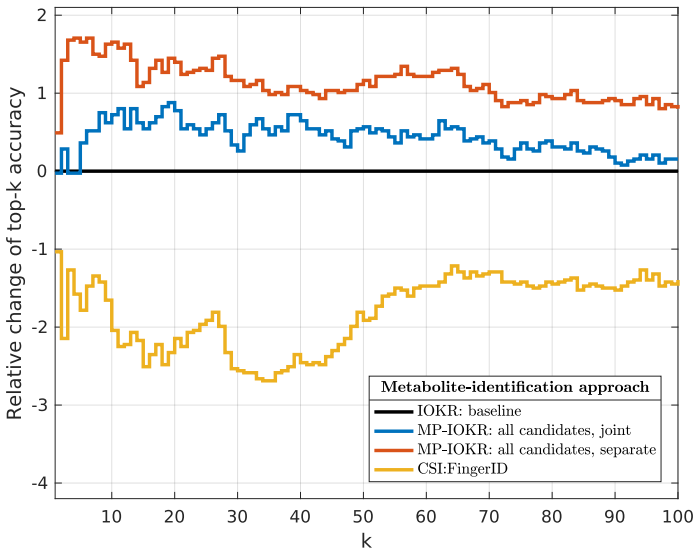

Figure 2: Difference of the top-k accuracy of the different metabolite identification approaches to the baseline method, which corresponds to IOKR.

also omitted CFM-ID as its performance is much lower than IOKR. The curves shows that MP-IOKR constantly outperforms IOKR and CSI:FingerID. Considering MP-IOKR, we observe that better performance is obtained when approximating the input feature vectors for the 24 input kernels separately rather than jointly. Using MP-IOKR with separate feature approximation, the top-1 accuracy can be improved by 0.5 point over IOKR and by up to 1.4 for the top-20. The accuracy improves the most in the top-10. This is an important measure in the metabolite identification task, as we cannot assume to reliably predict the true molecule at the top-1. In Table 4 we show that the ranks predicted using MP-IOKR are significantly better than with any other method in our comparison. We could not observe a significant difference between the different candidate selection strategies: using all the candidates for the magnitude preservation performs as well as using a random subset of candidates. We think this could be related to the fact, that in the solution of the MP-IOKR optimization problem in the case of approximated input features given in Equations (11) and respectively (13) the candidate output feature vectors appear as the candidate sets' mean vectors $\bar{\Psi}_C$ and covariance matrix $Cov\{\Psi_C\}$. This can be seen by developing the terms $\Psi' A^T$ and AA^T. We think that these statistics might not differ much between considering all candidates or a subset of them.

In Table 5 we compare the training and test running times for all methods. For the training step we fixed the hyperparameters and used 4138 training examples. In the test phase we identified the metabolites of 625 test examples. For MP-IOKR using separate input feature approximation we used 5 input kernels as this was the number of kernels with different γ_i's during our experiments. Thus in the training the input feature vectors of 5 kernels needed to be approximated. The calculation of the input kernels, fragmentation trees and molecular fingerprints were not taken into account. The reported running times

Table 4: P-values of the right-tailed sign test testing whether the ranks of the molecules predicted using MP-IOKR are significantly lower (better) than using one of the reference methods. All values are significant with $p < 0.0001$.

Method	**MP-IOKR**			
	all, joint	all, separated	random, joint	random, separated
CFM-ID	$1.05 \cdot 10^{-116}$	$1.48 \cdot 10^{-122}$	$2.15 \cdot 10^{-117}$	$1.68 \cdot 10^{-121}$
CSI:FingerID	$5.71 \cdot 10^{-11}$	$3.75 \cdot 10^{-13}$	$3.83 \cdot 10^{-12}$	$3.95 \cdot 10^{-13}$
IOKR	$3.10 \cdot 10^{-42}$	$2.14 \cdot 10^{-08}$	$5.37 \cdot 10^{-88}$	$6.54 \cdot 10^{-11}$

Table 5: Running time comparison: 4138 (3868 for CFM-ID) examples have been used in the training phase and 625 example during testing. For MP-IOKR the training procedure can be split into the calculation of the candidate statistics and model estimation, whereby the former step does not need to be repeated during hyperparameter selection. The left value results from using all the candidates and the right one from using 1% randomly chosen candidates. *(j)* indicates the joint input feature approximation and *(s)* the separate one.

Method	**Training time**		**Test time**
CFM-ID	870 h 11 min 18 s		3287 h 36 min 49 s
CSI:FingerID	82 h 28 min 23 s		1 h 11 min 31 s
IOKR	**10 s**		**2 min 16 s**
	candidate statistics	model estimation	
MP-IOKR (j)	1 h 42 min 8 s / 11 min 18 s	53 s	**2 min 17 s**
MP-IOKR (s)	1 h 42 min 8 s / 11 min 18 s	25 min 12 s	**2 min 17 s**

are given in CPU-time. The comparison shows that IOKR can be trained within a few seconds, while MP-IOKR takes from 12 minutes up to 2 hours (candidate statistics + model estimation) depending on the candidate selection and input feature approximation strategy. However, the calculation of the candidate statistics $\bar{\Psi}_C$ and $Cov\{\Psi_C\}$ (see previous paragraph) needs to be done only ones per training set. For the selection of the hyperparmaters, i.e. γ_i's and λ, only the model estimation step, i.e. a matrix inversion (see Equation (11) respectively (13)), needs to be repeated. The MP-IOKR training time is at least 40 times shorter than for CSI:FingerID and at least 400 times than for CFM-ID. In the testing time MP-IOKR and IOKR perform almost equally and outperform CSI:FingerID and CFM-ID clearly. For CFM-ID the simulation of the $514,483$ candidate spectra takes most of the time.

The complexity of the IOKR optimization is dominated by the inversion of an $\ell \times \ell$ matrix. Compared to that the complexity of MP-IOKR depends also on the number of candidates used in the magnitude preservation (see Section 2.3.1 and 2.3.2). This explains the increase of the training time from a few seconds to several minutes by using MP-IOKR over IOKR. It furthermore explains the training time difference between using all and a

random subset of candidates. The complexity of MP-IOKR further increases by using the separate input feature approximation, as the number of input kernels appears as factor in the complexity as well (see Section 2.3.2). It is worth mentioning that the training time can be significantly reduced by considering only a random subset of candidates without loss of identification performance.

4. Conclusions

In this paper, we propose a new method for structured prediction, based on combining the Input-Output Kernel Regression (IOKR) framework with magnitude-preserving ranking. The method, magnitude-preserving IOKR, is able to take advantage of the set of candidate outputs during the learning of the model, and thus leads to a better predictive performance than the pure regression approach. Interestingly, the magnitude-preserving extension turns out to correspond to IOKR with modified input and output features, by centering them within the candidate sets. For the frequent case of candidate sets with outputs lacking the corresponding 'correct' input, we derive an extension that approximates the inputs of the candidate sets. Our experiments confirm the benefits of magnitude-preserving IOKR.

Acknowledgments

This work has been supported by the Academy of Finland under the grants 268874 (MIDAS), 295496 (D4Health), and 310107 (MACOME). We acknowledge the computational resources provided by the Aalto Science-IT project.

References

F. Allen, R. Greiner, and D. Wishart. Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification. *Metabolomics*, 11:98–110, 2015.

S. Böcker and F. Rasche. Towards de novo identification of metabolites by analyzing tandem mass spectra. *Bioinfomatics*, 24(16):i49–i55, 2008.

C. Brouard, F. d'Alché-Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 593–600, 2011.

C. Brouard, H. Shen, K. Dührkop, F. d'Alché-Buc, S. Böcker, and J. Rousu. Fast metabolite identification with Input Output Kernel Regression. *Bioinformatics*, 32(12):i28–i36, 2016a.

C. Brouard, M. Szafranski, and F. d'Alché-Buc. Input Output Kernel Regression: supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17:1–48, 2016b.

C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 153–160, New York, NY, USA, 2005. ACM.

C. Cortes, M. Mohri, and A. Rastogi. Magnitude-preserving ranking algorithms. In *Proceedings of the 24th International Conference on Machine Learning*, pages 169–176. ACM, 2007.

C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(1):795–828, 2012.

K. Dührkop, H. Shen, M. Meusel, J. Rousu, and S. Böcker. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 2015.

P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Kernelizing the output of tree-based methods. In *Proceedings of the 23th International Conference on Machine Learning*, pages 345–352, 2006.

S. Giguère, A. Rolland, F. Laviolette, and M. Marchand. Algorithms for the hard pre-image problem of string kernels and the general problem of string prediction. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2021–2029, 2015.

H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 471–479, 2013.

S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic Acids Research*, 44(D1):D1202, 2016.

C. A. Micchelli and M. A. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Boberg, and T. Salakoski. Learning to rank with pairwise regularized least-squares. In T. Joachims, H. Li, T.-Y. Liu, and C. Zhai, editors, *SIGIR 2007 workshop on learning to rank for information retrieval*, volume 80, pages 27–33, 2007.

G. J. Patti, O. Yanes, and G. Siuzdak. Innovation: Metabolomics: the apogee of the omics trilogy. *Nature reviews Molecular Cell Biology*, 13(4):263–269, 2012.

G. Pedrick. Theory of reproducing kernels for Hilbert spaces of vector-valued functions. Technical report, University of Kansas, Department of Mathematics, 1957.

H. Shen, K. Dührkop, S. Böcker, and J. Rousu. Metabolite identification through multiple kernel learning on fragmentation trees. *Bioinformatics*, 30(12):i157–i164, 2014.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *International Conference on Language Resources and Evaluation (LREC)*, pages 2142–2147, 2006.

M. Wang, J. J. Carver, V. V. Phelan, L. M. Sanchez, N. Garg, Y. Peng, D. D. Nguyen, J. Watrous, C. A. Kapono, T. Luzzatto-Knaan, et al. Sharing and community curation of mass spectrometry data with global natural products social molecular networking. *Nature biotechnology*, 34(8):828–837, 2016.

J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik. Kernel dependency estimation. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.