Structured Output Prediction using Covariance-based Operator-valued Kernels

Hachem Kadri  
Mohammad Ghavamzadeh  
Philippe Preux  
INRIA Lille - Nord Europe, Team SequeL, France

Abstract

We study the problem of structured output learning from a regression perspective. We first provide a general formulation of the kernel dependency estimation (KDE) problem using operator-valued kernels. We then propose covariance-based operator-valued kernels that allow to take into account the structure of the kernel feature space. Finally, we evaluate the performance of our KDE approach using both covariance and conditional covariance kernels on two structured output problems, and compare it to the state-of-the-art kernel-based structured output regression methods.

1. Introduction

In many situations one is faced with the task of learning a mapping between objects of different nature, which can be generally characterized by complex data structures. This is often the case in a number of practical applications such as statistical machine translation (Wang & Shawe-Taylor, 2010), and speech recognition or synthesis (Cortes et al., 2005). Designing algorithms that are sensitive enough to detect structural dependencies among these complex data is becoming increasingly important. While classical learning algorithms can be easily extended to include complex inputs, more refined and sophisticated algorithms are needed to handle complex outputs. Recently, great efforts have been made to expand the applicability and robustness of general learning engines to structured output contexts. One difficulty encountered when working with structured data is that usual Euclidean methodologies cannot be applied in this case. Reproducing kernels provide an elegant way to overcome this problem. For structured outputs, two different but closely related kernel-based approaches can be found in the literature: kernel dependency estimation (KDE) (Weston et al.) and joint kernel maps (Tschantaridis et al., 2005; Weston et al., 2007). In this paper we will focus our attention on KDE-type methods for structured output learning. KDE is a regression based approach and we agree with Cortes et al. (2005) that minimizing a similarity measure capturing the closeness of the outputs in the output feature space (like in regression) is a natural way of learning the relevant structure. Moreover, avoiding the computation of pre-images during training is an important advantage over joint-kernel based methods. The main drawback of previous formulations of the KDE approach is that prior known input-output correlations cannot be easily incorporated into the method. We overcome this problem by generalizing these formulations and allowing KDE to capture output and input-output dependencies using operator-valued (multi-task) kernels. Our idea is to improve the high-dimensional regression step in KDE by doing an operator-valued KRR rather than a scalar-valued one. By this way, operator-valued kernels allow to map input data and structured output features into a joint feature space in which existing correlations can be exploited. It should be pointed out that operator-valued kernels have been recently used for structured outputs in order to deal with the problem of link prediction (Brouard et al., 2011). The authors showed that their operator-valued kernel extension of output kernel regression fits the first step of KDE. However, they only considered the case of identity-based operator-valued kernels, and did not discuss the associated pre-image problem since their link prediction application did not require this step.
2. Operator-valued Kernel Formulation of Kernel Dependency Estimation

Given \((x_i, y_i)_{i=1}^n \in X \times Y\) where \(X\) is the input space and \(Y\) the structured output space, we consider the problem of learning the mapping \(f\) from \(X\) to \(Y\). The idea of KDE is to embed output data by using a mapping \(\Phi_1\) between the structured output space \(Y\) and a Euclidean feature space \(F_Y\) defined by a scalar-valued a kernel \(l\). Instead of learning \(f\) in order to predict an output \(y\) for an input \(x\), we first learn the mapping \(g\) from \(X\) to \(F_Y\), and then compute the pre-image of \(g(x)\) by the inverse mapping of \(\Phi_l\), i.e., \(y = f(x) = \Phi_l^{-1}(g(x))\) (see Figure 1). Note that in the original KDE algorithm (Weston et al.), the feature space \(F_Y\) is reduced using kernel PCA. However, all previous KDE formulations use ridge regression with a scalar-valued kernel \(k\) on \(X\) to learn the mapping \(g\). This approach has the drawback of not taking into account dependencies between data in the feature space \(F_Y\). To overcome this problem, we propose to use an operator-valued (multi-task) kernel-based regression approach to learn \(g\) in a vector (or function)-valued RKHS built from a kernel which outputs an operator that encodes the relationship between the output’s components. As we show in Figure 1, the feature space introduced by an operator-valued kernel is a joint feature space which contains information of the input space \(X\) and the output feature space \(F_Y\).

We now describe our KDE formulation that takes into account the general case where the feature spaces associated to input and output kernels can be infinite dimensional. We start by recalling some properties of these kernels and their associated RKHSs. For more details in the context of learning theory, see Micchelli & Pontil (2005). Let \(L(F_Y)\) the set of bounded operators from \(F_Y\) to \(F_Y\).

**Definition 1 (Non-negative \(L(F_Y)\)-valued kernel)**
A non-negative \(L(F_Y)\)-valued kernel \(K\) is an operator-valued function on \(X \times X\), i.e., \(K : X \times X \rightarrow L(F_Y)\), such that for all \(x_i, x_j \in X\), \(\varphi_i, \varphi_j \in F_Y\), and \(m \in \mathbb{N}^+\)

i. \(K(x_i, x_j) = K(x_j, x_i)^\ast\) (* denotes the adjoint),

ii. \(\sum_{i,j=1}^m \langle K(x_i, x_j) \varphi_i, \varphi_j \rangle_{F_Y} \geq 0\).

Given a non-negative \(L(F_Y)\)-valued kernel \(K\) on \(X \times X\), there exists a unique RKHS of \(F_Y\)-valued functions defined over \(X \times X\) whose reproducing kernel is \(K\).

**Definition 2 (\(F_Y\)-valued RKHS)**
A RKHS \(F_{XY}\) of \(F_Y\)-valued functions \(g : X \rightarrow F_Y\) is a Hilbert space such that there is a non-negative \(L(F_Y)\)-valued kernel \(K\) with the following properties:

\[ \arg \min_{g \in F_{XY}} \sum_{i=1}^n \|g(x_i) - \Phi_l(y_i)\|_{F_Y}^2 + \lambda \|g\|^2, \]  \hfill (1)

where \(\lambda > 0\) is a regularization parameter. Using the representer theorem, the solution of (1) has the following form:

\[ g(\cdot) = \sum_{i=1}^n K(\cdot, x_i) \psi_i \]  \hfill (2)

where \(\psi_i \in F_Y\). Substituting (2) in (1), we obtain an analytic solution for the optimization problem as

\[ \Psi = (K + \lambda I)^{-1} \Phi_1, \]  \hfill (3)

where \(\Phi_1\) is the column vector of \([\Phi_l(y_i) \in F_Y]_{i=1}^n\). Note that features \(\Phi_l(y_i)\) in this equation can be ex-
plicit or implicit. We show in the following by substituting (3) in the next step (step 2) that, even with implicit features, we are still able to formulate the structured output prediction problem in terms of explicit quantities that are readily computable via input and output kernels.

**Step 2 (pre-image) Prediction:** In order to compute the structured prediction \( f(x) \) for an input \( x \), we solve the following pre-image problem:

\[
\begin{align*}
    f(x) &= \arg \min_{y \in \mathcal{Y}} \| y(x) - \Phi_1(y) \|^2_{\mathcal{F}_Y} \\
    &= \arg \min_{y \in \mathcal{Y}} \| \sum_{i=1}^{n} K(x, x) \psi_i - \Phi_1(y) \|^2_{\mathcal{F}_Y} \\
    &= \arg \min_{y \in \mathcal{Y}} \| K_x \Psi - \Phi_1(y) \|^2_{\mathcal{F}_Y} \\
    &= \arg \min_{y \in \mathcal{Y}} \| K_x (K + \lambda I)^{-1} \Phi_1 - \Phi_1(y) \|^2_{\mathcal{F}_Y} \\
    &= \arg \min_{y \in \mathcal{Y}} \| l(y - 2(K_x (K + \lambda I)^{-1} \Phi_1, \Phi_1(y))_{\mathcal{F}_Y},
\end{align*}
\]

where \( K_x \) is the row vector of operators corresponding to the input \( x \). At this step, the operator-valued kernel formulation has an inherent difficulty in expressing the pre-image problem without kernel mappings. Indeed, \( \Phi_1 \) can be unknown and only implicitly defined through the kernel \( l \). In this case, the usual kernel trick is not sufficient to solve the pre-image problem. Thus, we introduce the following variant (generalization) of the kernel trick: \( \langle T \Phi_1(y_1), \Phi_1(y_2) \rangle_{\mathcal{F}_Y} = \langle Tl(y_1, \cdot), (y_2) \rangle \), where \( T \) is an operator in \( \mathcal{L}(\mathcal{F}_Y) \). Note that the usual kernel trick \( \langle \Phi_1(y_1), \Phi_1(y_2) \rangle_{\mathcal{F}_Y} = l(y_1, y_2) \) is recovered when \( T \) is the identity operator. It is easy to check that our proposed trick holds if we consider the feature space associated to the kernel \( l \), i.e., \( \Phi_i(y) = l(y, \cdot) \).

Using this trick, the pre-image problem can be written as

\[
    f(x) = \arg \min_{y \in \mathcal{Y}} l(y - 2(K_x (K + \lambda I)^{-1} \Phi_1, \Phi_1(y))_{\mathcal{F}_Y},
\]

where \( L_x \) is the column vector whose \( i \)-th element is \( l(y_i, \cdot) \). We can now express \( f(x) \) only using kernel functions.

### 3. Covariance-based Operator-valued Kernels

In this section, we study the problem of designing operator-valued kernels suitable for structured outputs in the KDE formulation. This is quite important in order to take full advantage of the operator-valued KDE formulation. In fact, the main purpose of using the operator-valued kernel formulation is to take into account the dependencies between the variables \( \Phi_i(y_i) \), i.e., the projection of \( y_i \) in the feature space \( \mathcal{F}_Y \), with the objective of capturing the structure of the output data encapsulated in \( \Phi_1(y_i) \). Since \( \mathcal{F}_Y \) is a RKHS, we focus our attention to operators that act on scalar-valued RKHSs. Covariance operators on RKHS have recently received a considerable amount of attention in the machine learning community. These operators that provide the simplest measure of dependency have been successfully applied to the problem of dimensionality reduction (Fukumizu et al., 2004). We propose to use the following covariance-based operator-valued kernel in our KDE formulation:

\[
    K(x_i, x_j) = k(x_i, x_j)C_{YY},
\]

where \( k \) is a scalar-valued kernel on \( \mathcal{X} \) and \( C_{YY} : \mathcal{F}_Y \to \mathcal{F}_Y \) is the covariance operator defined as

\[
    \langle \varphi_i, C_{YY} \varphi_j \rangle_{\mathcal{F}_Y} = E[\varphi_i(\mathcal{Y}) \varphi_j(\mathcal{Y})].
\]

The empirical covariance operator \( \hat{C}^{(n)}_{YY} \) is given by

\[
    \hat{C}^{(n)}_{YY} = \frac{1}{n} \sum_{i=1}^{n} l(\cdot, y_i) \otimes l(\cdot, y_i),
\]

where \( \otimes \) is the tensor product \( \langle \varphi_1 \otimes \varphi_2 \rangle h = \langle \varphi_2, h \rangle \varphi_1 \).

The operator-valued kernel in Eq. (5) is a separable kernel since it operates on the output space and then encodes the interactions between the outputs without any reference to the input space. To address this issue, we propose a variant of this kernel based on the conditional covariance operator rather than the covariance operator and define it as follows:

\[
    K(x_i, x_j) = k(x_i, x_j)C_{YY | X},
\]

where \( C_{YY | X} = C_{YY} - C_{YX} C_{XX}^{-1} C_{XY} \) is the conditional covariance operator on \( \mathcal{F}_Y \). This operator allows the operator-valued kernel to simultaneously encode the correlations between the outputs and to take into account (non-parametrically) the effects of the input data. In Proposition 1, we show how the pre-image problem (4) can be formulated using the covariance-based operator-valued kernels in Eqs. (5) and (7).

**Proposition 1** The Gram matrix expression of the pre-image problem in Eq. (4) can be written for the case of the covariance operator-valued kernels, Eq. (5), and conditional covariance operator-valued kernels, Eq. (7), as follows:

\[
    \arg \min_{y \in \mathcal{Y}} l(y - 2L_y^T (k_x \otimes T)(k \otimes T + n \lambda I_n)^{-1} \text{vec}(I_n),
\]

where \( T = L \) for the covariance operator and \( T = L - (k + n \epsilon I_n)^{-1} k L \) for the conditional covariance operator in which \( \epsilon \) is a regularization parameter required for...
for the operator inversion, \( k \) and \( \mathbf{L} \) are Gram matrices associated to the scalar-valued kernels \( k \) and \( l \), \( k_x \) and \( \mathbf{L}_y \) are the column vectors \( (k(x,x_1), \ldots, k(x,x_n))^\top \) and \( (l(y,y_1), \ldots, l(y,y_n))^\top \), \( \text{vec} \) is the vector operator such that \( \text{vec}(A) \) is the vector of columns of the matrix \( A \), and finally \( \otimes \) is the Kronecker product.

### 4. Experimental Results

In this section, we evaluate our operator-valued kernel formulation on an image reconstruction problem Weston et al.. This problem takes the top half (the first 8 pixel lines) of a USPS postal digit as input and estimates its bottom half. We apply our KDE method using both covariance and conditional covariance operator-valued kernels and compare it with the KDE algorithms of Weston et al. and Cortes et al. (2005). In all these methods, we use RBF kernels for both input and output with the parameters shown in Table 1. This table also contains the ridge parameter used by these algorithms. We tried a number of values for these parameters and those in the table yielded the best performance.

We perform 5-fold cross validation on 600 digits selected randomly from the first 1000 digits of the USPS handwritten 16 by 16 pixel digit database, training with a single fold (200 examples as in Weston et al.) and testing on the remainder. Given a test input, we solve the problem and then choose as output the pre-image from the training data that is closest to this solution. The loss function used to evaluate the prediction \( \hat{y} \) for an output \( y \) is the RBF loss induced by the output kernel, i.e., \( || \Phi(y) - \Phi(\hat{y}) ||^2 = 2 - 2 \exp \left( - ||y - \hat{y}||^2 / (2\sigma^2) \right) \). Table 1 shows the mean and standard deviation of the RBF loss and the kernel and ridge parameters for the four KDE algorithms described above.

Our proposed operator-valued kernel approach produced promising results in this experiment. Improvements in prediction accuracy obtained over the previous KDE algorithms can be explained by the fact that using the covariance operator we can capture the output correlations, and using the conditional covariance we can take also into account information from the inputs. In this context, kPCA-based KDE (the algorithm by Weston et al.) performs better than the KDE formulation of Cortes et al. (2005). Indeed, the KDE formulation of Cortes et al. (2005) is equivalent to using an identity-based operator-valued kernel in our formulation, and thus, incapable in considering dependencies in the output feature space (contrary to the other methods considered here).

### 5. Conclusion

In this paper, we studied the problem of structured output learning using kernel dependency estimation methods. We provided a general formulation of this problem using operator-valued kernels. We then defined a covariance-based operator-valued kernel for our proposed KDE formulation. This kernel takes into account the structure of the output kernel feature space and encodes the interactions between the outputs, but makes no reference to the input space. We addressed this issue by introducing a variant of our KDE method based on the conditional covariance operator that in addition to the correlation between the outputs takes into account the effects of the input variables. Finally, we evaluated the performance of our KDE method using both covariance and conditional covariance kernels on an image reconstruction problem.

### References


1These results are obtained using the Spider toolbox available at www.kyb.mpg.de/bs/people/spider.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RBF Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDE - Cortes</td>
<td>1.0423 ± 0.0996</td>
</tr>
<tr>
<td>KDE - Weston</td>
<td>1.0246 ± 0.0529</td>
</tr>
<tr>
<td>KDE - Covariance</td>
<td>0.7616 ± 0.0304</td>
</tr>
<tr>
<td>KDE - Conditional Covariance</td>
<td>0.6241 ± 0.0411</td>
</tr>
</tbody>
</table>

Table 1. Performance of the KDE algorithms on an image reconstruction problem of handwritten digits.