Semi-supervised Penalized Output Kernel Regression for Link Prediction

Céline Brouard CELINE.BROUARD@IBISC.UNIV-EVRY.FR
Florence d'Alché-Buc FLORENCE.DALCHE@IBISC.UNIV-EVRY.FR
IBISC, EA 4526, Université d'Évry Val d'Essonne, F–91025 Évry cedex, France

Marie Szafranski MARIE.SZAFRANSKI@IBISC.UNIV-EVRY.FR
ENSIIE, F–91025 Évry cedex, France
IBISC, EA 4526, Université d'Évry Val d'Essonne, F–91025 Évry cedex, France

Abstract

Link prediction is addressed as an output kernel learning task through semi-supervised Output Kernel Regression. Working in the framework of RKHS theory with vector-valued functions, we establish a new representer theorem devoted to semi-supervised least square regression. We then apply it to get a new model (POKR: Penalized Output Kernel Regression) and show its relevance using numerical experiments on artificial networks and two real applications using a very low percentage of labeled data in a transductive setting.

1. Introduction

Recent years have witnessed a surge of interest for network inference in social networks as well as in biological networks. Link prediction (Huynen et al., 2003; Liben-Nowell & Kleinberg, 2007), defined as a supervised task, aims at building pairwise classifiers able to predict if two components interact, from a dataset of labeled pairs of components. The underlying hypothesis is that some input features relative to each node in a pair provide valuable information about the presence or the absence of a link. The main approaches devoted to this task fall into two families: probabilistic graphical models (Miller et al., 2009; Taskar et al., 2003) provide posterior probabilities of links while kernel or similarity based methods take benefit from the versatility of kernels to encode various structured knowledge in the input space as well as in the output space (Yamanishi et al., 2004; Ben-Hur & Noble, 2005; Geurts et al., 2006; 2007a;b).

However in many fields, there exists additional information about the nodes, even if we don't know their interactions. For instance, in biology, it is much easier to get a detailed description of the properties of a protein compared to the cost of experimental methods used to detect physical interactions between two proteins. The question of semi-supervised link prediction is thus really meaningful and curiously has not been well explored in the literature. Moreover, in some cases, we already know the finite set of candidate nodes in the target graph and the problem can therefore be stated as a transductive one: we want to complete a partially known network we have at hand.

The aim of this paper is to develop methods that exploit unlabeled data. To address this issue, we have chosen to convert the binary pairwise classification problem into an output kernel learning problem, as in (Geurts et al., 2006; 2007b). A target output kernel is assumed to encode the similarity of data as nodes in the graph and the goal of learning is to approximate this function by using appropriate input features. Using the kernel trick in the output space allows one to reduce the problem of learning from pairs to learning a single variable function with values in the output feature space. This supervised regression task is referred as Output Kernel Regression (OKR). Once the output kernel is learnt, a link prediction is performed by thresholding the kernel value for a pair of inputs. Tree-based methods have been developed to solve OKR and have been applied to supervised biological networks inference (Geurts et al., 2007a).

To benefit from the usually large amount of unlabeled data, we need to extend OKR to semi-supervised learning. A powerful approach to semi-supervised re-

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

gression is based on graph-based regularization that forces the prediction function to be smooth on the graph describing similarities between inputs. Enforcing smoothness of the function permits to propagate output labels over close inputs as shown in (Zhou et al., 2004; Belkin & Niyogi, 2004). Belkin et al. (2006) have proposed to explicitly embed such ideas into the framework of regularization within Reproducing Kernel Hilbert Space (RKHS) for real-valued functions. This allows one to benefit from a representer theorem devoted to semi-supervised learning that provides a basis for new models and algorithms.

Exploiting this regularization framework in the case of OKR requires to define an appropriate input kernel in the proper RKHS theory. Here, the function to be learnt is not real-valued but vector-valued in a Hilbert space. We therefore need to turn to the RKHS theory, devoted to vector-valued functions (Senkene & Tempel'man, 1973; Micchelli & Pontil, 2005). In this theory, kernels are operator-valued and applied to vectors of the given Hilbert space. In particular, similarly to the theory in the scalar case, a RKHS can be build from a positive definite kernel and representer theorems can be proven (Micchelli & Pontil, 2005). While being very powerful, this theory is still underused. We must emphasize the series of works (Caponnetto et al., 2008; Argyriou et al., 2009) that developed this theory to solve a multi-tasks learning problem and the recent work of Kadri et al. (2010) who achieved functional regression by applying this theory for input and output functional space.

In this work, the RKHS theory with vector-valued functions provides us with a general framework for OKR. Starting from the results existing in the supervised case for Tikhonov regularization (Micchelli & Pontil, 2005), we show that with an appropriate choice of the operator-valued kernel based on some input scalar kernel, we directly retrieve the extension of kernel ridge regression to output kernels proposed by Cortes et al. (2005). We propose a new representer theorem devoted to semi-supervised learning that leads us to define a new model, expressed as a closed-form solution. While the development of the approach overcomes the link prediction problem, we use the obtained model to approximate the output kernel on several tasks: a first set of artificial networks, the NIPS co-authorship network and the well known yeast protein-protein interaction network dataset.

In the rest of this paper, we first introduce the existing framework of OKR for link prediction. In section 3, we briefly recall the RKHS theory devoted to functions with values in a Hilbert space. Section 4 is devoted to supervised learning and the derivation of a closed-form solution in the case of penalized least square cost by choosing an adequate operator-valued input kernel. Section 5 presents the core results of the paper: a new representer theorem devoted to semi-supervised learning in the case of vector-valued functions and a resulting new model, still expressed in a closed-form. In section 6, we present experimental results in a transductive setting and in section 7, we draw some conclusions and perspectives.

2. Link prediction with Output Kernel Regression

Let us define $\mathcal{O}$ the set of descriptions of the objects (individuals, proteins, authors) we are interested in. Assume that there exists some relation $f_{target} : \mathcal{O} \times \mathcal{O} \to \{0, 1\}$ that we want to approximate. f_{target} could be social relationships like friendship or co-authorship if the nodes are individuals, or physical interactions if nodes are proteins. During the training phase we are given $\mathcal{G}_\ell = (\mathcal{O}_\ell, A_\ell)$, a non oriented graph defined by the subset $\mathcal{O}_\ell \subseteq \mathcal{O}$ and the adjacency matrix A_ℓ of size $\ell \times \ell$ such that $A_\ell(i, j) = f_{target}(o_i, o_j)$. Supervised link prediction consists of learning a binary pairwise classifier $f : \mathcal{O} \times \mathcal{O} \to \{0, 1\}$ that predicts if two objects interact or not, from the training information $\mathcal{G}_\ell$.

In this work, we convert the binary pairwise classification task into an output kernel learning task as in (Geurts et al., 2006). This is made possible by noticing that a Gram matrix K_{Y_ℓ} can be defined from the adjacency matrix A_ℓ using any kernel that encodes the proximities of nodes in a given graph. Typically, we use in this work the diffusion kernel (Kondor & Lafferty, 2002) matrix $K_{Y_\ell} = \exp(-\beta L_{Y_\ell})$ where $L_{Y_\ell} = D_\ell - A_\ell$, with D_ℓ the degree matrix.

In the training information, the matrix A_ℓ is now replaced by a positive semi-definite matrix K_{Y_ℓ}. We assume that a positive definite kernel κ_y: $\mathcal{O} \times \mathcal{O} \to \mathbb{R}$ underlies this Gram matrix such that $\forall i, j \leq \ell, K_{Y_\ell}(i, j) = \kappa_y(o_i, o_j)$. Moreover, there exists an Hilbert space $\mathcal{F}_y$, called the feature space, and a feature map $y : \mathcal{O} \to \mathcal{F}_y$ such that $\forall (o, o') \in \mathcal{O}, \kappa_y(o, o') = \langle y(o), y(o') \rangle_{\mathcal{F}_y}$.

The idea underlying output kernel learning is the following: we assume that an approximation of the kernel function κ_y will provide valuable information about the proximity of the objects of $\mathcal{O}$ as nodes in the unknown whole graph defined on $\mathcal{O}$. Given that assumption, a classifier f_θ is defined from the approximation $\widehat{\kappa_y}$ by thresholding its output values:

$$f_\theta(o, o') = \mathrm{sgn}(\widehat{\kappa_y}(o, o') - \theta) .$$

We build an approximation of κ_y from the inner product between the outputs of a single variable function $h : \mathcal{O} \to \mathcal{F}_y$: $\widehat{\kappa_y}(o, o') = \langle h(o), h(o') \rangle_{\mathcal{F}_y}$. By this way, the approximation $\widehat{\kappa_y}$ is a kernel by construction and there is no need to solve a pre-image problem. Instead of learning a pairwise classifier, we need to learn a single variable function with output values in a Hilbert space (the output feature space $\mathcal{F}_y$). In the following, we show how the RKHS theory devoted to vector-valued functions can provide a theoretical framework both in the supervised and semi-supervised cases.

3. Supervised Input Output Kernel Regression with RKHS theory

The RKHS theory for real valued functions provides a powerful theoretical framework for regularization. Numerous models including ridge regression and support vector machines can be derived from the application of the representer theorem and different data-dependent cost functions.

In the following, we briefly recall the main elements of the RKHS theory that we need for regularization of vector-valued functions. We especially focus on the penalization of the least square cost in order to benefit from the kernel trick in the output space.

For a given Hilbert space $\mathcal{F}_y$, we note $\mathcal{L}(\mathcal{F}_y)$, the set of all bounded linear operators from $\mathcal{F}_y$ to itself. Given $A \in \mathcal{L}(\mathcal{F}_y)$, A^* denotes the adjoint of A.

Definition 1 (Operator-valued kernel (Senkene & Tempel'man, 1973; Caponnetto et al., 2008))**.** Let $\mathcal{O}$ be a set and $\mathcal{F}_y$ an Hilbert space. $\mathcal{K}_x : \mathcal{O} \times \mathcal{O} \to \mathcal{L}(\mathcal{F}_y)$ is a kernel if:

- $\forall (o, o') \in \mathcal{O} \times \mathcal{O}$, $\mathcal{K}_x(o, o') = \mathcal{K}_x(o, o')^*$
- $\forall m \in \mathbb{N}$, $\forall \{(o_i, \mathbf{y}_i)\}_{i=1}^m \subseteq \mathcal{O} \times \mathcal{F}_y$,

$$\sum_{j=1}^{m} \langle \mathbf{y}_i, \mathcal{K}_x(o_i, o_j)\mathbf{y}_j \rangle_{\mathcal{F}_y} \geq 0 \ .$$

The following theorem says that one can build a RKHS from a given operator-valued kernel.

Theorem 2 (Senkene & Tempel'man (1973); Micchelli & Pontil (2005))**.** *Let $\mathcal{O}$ be a set and $\mathcal{F}_y$ be an Hilbert space. If $\mathcal{K}_x : \mathcal{O} \times \mathcal{O} \to \mathcal{L}(\mathcal{F}_y) \in \mathcal{F}_y$ is an operator-valued kernel, then there exists a unique RKHS $\mathcal{H}_{\mathcal{K}_x}$ which admits $\mathcal{K}_x$ as the reproducing kernel, that is*

$$\forall o \in \mathcal{O}, \forall \mathbf{y} \in \mathcal{F}_y, \ \langle h, \mathcal{K}_x(o, \cdot)\mathbf{y} \rangle_{\mathcal{H}} = \langle h(o), \mathbf{y} \rangle_{\mathcal{F}_y} \ . \quad (1)$$

For sake of simplicity we omit $\mathcal{K}_x$ and use $\mathcal{H} = \mathcal{H}_{\mathcal{K}_x}$ in the rest of the paper. As in the scalar case, one of the most appealing feature of RKHS is to provide a theoretical framework for regularization, the so-called representer theorems. We focus here on the representer theorem devoted to penalized least square.

Theorem 3 (Micchelli & Pontil (2005))**.** *Let $\mathcal{O}$ be a set and $\mathcal{F}_y$ be an Hilbert space. Given a set of labeled examples $S_\ell = \{(o_i, \mathbf{y}_i)\}_{i=1}^\ell \subseteq \mathcal{O} \times \mathcal{F}_y$, a RKHS $\mathcal{H}$ with reproducing kernel $\mathcal{K}_x : \mathcal{O} \times \mathcal{O} \to \mathcal{L}(\mathcal{F}_y)$, the minimizer $\hat{h}$ of the following optimization problem:*

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} J(h) = \sum_{i=1}^{\ell} \|h(o_i) - \mathbf{y}_i\|_{\mathcal{F}_y}^2 + \lambda_1 \|h\|_{\mathcal{H}}^2 \ , \quad (2)$$

with $\lambda_1 > 0$, admits an expansion:

$$\hat{h}(\cdot) = \sum_{j=1}^{\ell} \mathcal{K}_x(o_j, \cdot)\mathbf{c}_j \ , \quad (3)$$

where the vectors $\mathbf{c}_j \in \mathcal{F}_y, j = \{1, \cdots, \ell\}$ satisfy the equations:

$$\mathbf{y}_j = \sum_{i=1}^{\ell} (\mathcal{K}_x(o_i, o_j) + \lambda_1 \delta_{ij})\mathbf{c}_i \ , \quad (4)$$

where δ is the Kronecker symbol: $\delta_{ii} = 1$ and $\forall j \neq i$, $\delta_{ij} = 0$.

To benefit from the RKHS theory, we must define a suitable input operator-valued kernel in the context of OKR. In the following, we slightly change OKR by assuming that an input (scalar) kernel is given and then we define a simple operator-valued kernel.

4. Input Output Kernel Regression within the appropriate RKHS theory

4.1. Scalar input kernel

OKR is extended to data described by some input kernel. The training input set is now defined by an input Gram matrix K_{X_ℓ}, which encodes for the properties of the training objects $\mathcal{O}_\ell$. As in the output case, the coefficients of the Gram matrix are supposed to be defined from a positive definite input kernel function $\kappa_x : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$, with $\forall i, j \leq \ell, K_{X_{\ell}(i,j)} = \kappa_x(o_i, o_j)$. Given κ_x, there exists an Hilbert space $\mathcal{F}_x$ and a feature map $x : \mathcal{O} \to \mathcal{F}_x$, such that $\forall (o, o') \in \mathcal{O} \times \mathcal{O}$, we have $\kappa_x(o, o') = \langle x(o), x(o') \rangle_{\mathcal{F}_x}$. But contrary to the output case, the input kernel function κ_x is assumed to be known, which is useful to make predictions on new data. We note that this extension of OKR directly fits the first step of Kernel Dependency Estimation (KDE) reformulation of Cortes et al. (2005).[1]

[1] The second step of KDE being the pre-image problem.

4.2. Operator-valued kernel

We define $\mathcal{K}_x$ as follows:

$$\begin{aligned} \mathcal{K}_x : \quad & \mathcal{O} \times \mathcal{O} \to \mathcal{L}(\mathcal{F}_y) \\ & (o, o') \mapsto \kappa_x(o, o') \times I_{\mathcal{F}_y} , \end{aligned} \tag{5}$$

with $I_{\mathcal{F}_y}$, the identity matrix of size $dim(\mathcal{F}_y)$.

We can briefly show that the kernel $\mathcal{K}_x$ satisfies the properties of a nonnegative kernel: it is symmetric thus hermitian. Moreover the positive semi-definite property of κ_x leads to $\mathcal{K}_x$ being positive definite: $\forall m \in \mathbb{N}$, $\forall \{(o_i, \mathbf{y}_i)\}_{i=1}^m \subseteq \mathcal{O} \times \mathcal{F}_y$,

$$\sum_{i,j=1}^{m} \langle \mathbf{y}_i, \mathcal{K}_x(o_i, o_j)\mathbf{y}_j \rangle_{\mathcal{F}_y} = \sum_{i,j=1}^{m} \kappa_x(o_i, o_j) \langle \mathbf{y}_i, \mathbf{y}_j \rangle_{\mathcal{F}_y} \geq 0.$$

Given that an operator-valued kernel is defined from (5), Theorem 2 ensures that a RKHS, $\mathcal{H}_{\mathcal{K}_x}$ can be built from it. The way to build $\mathcal{H}_{\mathcal{K}_x}$ is detailed in the proof of Theorem 2 given in (Senkene & Tempel'man, 1973; Micchelli & Pontil, 2005). Theorem 3 can then be applied and leads to the following closed-form solution:

Proposition 4. When $\mathcal{K}_x$ is defined by mapping (5), the solution of Problem (2) reads

$$C = Y_\ell (K_{X_\ell} + \lambda_1 I_\ell)^{-1} ,$$

where $Y_\ell = (\mathbf{y}_1^T, \cdots, \mathbf{y}_\ell^T)$ is a matrix of dimension $dim(\mathcal{F}_y) \times \ell$. K_{X_ℓ} is the Gram matrix of size $\ell \times \ell$ associated to kernel κ_x. Finally, I_ℓ is the identity matrix of size ℓ.

Proposition 4 provides thus for the h model:

$$\forall o \in \mathcal{O}, h(o) = C X_\ell^T x(o) ,$$

where $X_\ell = (x(o_1)^T, \cdots, x(o_\ell)^T)$ denotes a matrix of dimension $dim(\mathcal{F}_x) \times \ell$. It is worth noting that Theorem 3 and Proposition 4 provide a principled way to retrieve the linear model proposed by Cortes et al. (2005) in the framework of the reformulation of KDE.

5. Semi-supervised Output Kernel Regression

In the case of real-valued functions, Belkin et al. have introduced new representer theorem devoted to semi-supervised learning (2006, Theorem 2). A graph-based regularizer is added to the cost functional to enforce the smoothness of the target function h. We define the Laplacian L, of dimension $(\ell + u) \times (\ell + u)$, given by $L = D - W$, where W is a matrix measuring the similarity of objects in the input space[2] and the general term of the diagonal matrix D is $D_{ii} = \sum_{j=1}^{\ell+u} W_{ij}$.

On the following, we state and prove a representer theorem devoted to semi-supervised learning in the context of RKHS with functions in a Hilbert space. Indeed, Theorem 5 extends to vector-valued functions the representer theorem proposed by Belkin et al. (2006) in the scalar case. Besides, it also extends Theorem 3 to the semi-supervised framework.

Theorem 5. *Let $\mathcal{O}$ be a set and $\mathcal{F}_y$ an Hilbert space. Given a set of labeled examples $S_\ell = \{(o_i, \mathbf{y}_i)\}_{i=1}^{\ell} \subseteq \mathcal{O} \times \mathcal{F}_y$, a set of unlabeled examples $S_u = \{o_i\}_{i=\ell+1}^{\ell+u} \subseteq \mathcal{O}$, a RKHS $\mathcal{H}$ with reproducing kernel $\mathcal{K}_x : \mathcal{O} \times \mathcal{O} \to \mathcal{L}(\mathcal{F}_y)$, and a matrix W with positive values measuring the similarity of objects in the input space, the minimizer $\hat{h}$ of the following optimization problem:*

$$\operatorname*{argmin}_{h \in \mathcal{H}} J(h) = \sum_{i=1}^{\ell} \|h(o_i) - \mathbf{y}_i\|_{\mathcal{F}_y}^2 + \lambda_1 \|h\|_{\mathcal{H}}^2 \tag{6a}$$

$$+ \lambda_2 \sum_{i,j=1}^{\ell+u} W_{ij} \|h(o_i) - h(o_j)\|_{\mathcal{F}_y}^2 , \tag{6b}$$

with λ_1 and $\lambda_2 > 0$, admits an expansion:

$$\hat{h}(\cdot) = \sum_{j=1}^{\ell+u} \mathcal{K}_x(o_j, \cdot)\mathbf{c}_j , \tag{7}$$

where the vectors $\mathbf{c}_j \in \mathcal{F}_y, j = \{1, \cdots, (\ell + u)\}$ satisfy the equations:

$$V_j \mathbf{y}_j = V_j \sum_{i=1}^{\ell+u} \mathcal{K}_x(o_i, o_j)\mathbf{c}_i \; + \; \lambda_1 \mathbf{c}_j \tag{8a}$$

$$+ \; 2\lambda_2 \sum_{i=1}^{\ell+u} L_{ij} \sum_{m=1}^{\ell+u} \mathcal{K}_x(o_m, o_i)\mathbf{c}_m , \tag{8b}$$

where the matrix V_j of dimension $dim(\mathcal{F}_y) \times dim(\mathcal{F}_y)$ is the identity matrix if $j \leq \ell$ and the null matrix if $\ell < j \leq (\ell + u)$.

Sketch of the proof. Problem (6) admits a unique solution $\hat{h}$ given by (7), if for any $h \in \mathcal{H}$, $J(h) > J(\hat{h})$. To show this, we define $g = h - \hat{h}$ and establish that $J(h) = J(g + \hat{h}) = J(\hat{h}) + c$, with $c > 0$. The complete proof is provided in the supplementary material. [3] □

Now, if we use the operator-valued $\mathcal{K}_x$ defined by (5), we can again apply theorem 2 and build the corresponding RKHS. Now when integrating the definition

[2] In particular, one can choose $W_{ij} = \kappa_x(o_i, o_j)$.

[3] http://amis-group.fr/?q=supp_downloads

(5) into condition (8) of Theorem 5, we get the following closed-form solution:

Proposition 6. When $\mathcal{K}_x$ is defined by mapping (5), the solution of Problem (6) reads

$$C = Y_\ell U (K_{X_{\ell+u}} U^T U + \lambda_1 I_{\ell+u} + 2\lambda_2 K_{X_{\ell+u}} L)^{-1}, \quad (9)$$

where C is a matrix of dimension $\dim(\mathcal{F}_y) \times (\ell + u)$ that gather the vectors $\mathbf{c}_j$ of the expansion (7), $Y_\ell = (\mathbf{y}_1^T, \cdots, \mathbf{y}_\ell^T)$ is a matrix of dimension $dim(\mathcal{F}_y) \times \ell$. U denotes a matrix of dimension $\ell \times (\ell + u)$ that contains an identity matrix of size $\ell \times \ell$ on the left hand side and a zero matrix of size $\ell \times u$ on the right hand side. $K_{X_{\ell+u}}$ is the Gram matrix of size $(\ell+u)\times(\ell+u)$ associated to kernel κ_x. Finally, $I_{\ell+u}$ is the identity matrix of size $(\ell + u)$.

Thus, Equation (9) provides for the model h:

$$\forall o \in \mathcal{O},\ h(o) = C X_{\ell+u}^T x(o)\ ,$$

where $X_{\ell+u} = (x(o_1)^T, \cdots, x(o_{\ell+u})^T)$ denotes a matrix of dimension $dim(\mathcal{F}_x) \times (\ell + u)$. Hence, the computation of this solution mainly requires the inversion of a matrix of size $(\ell + u) \times (\ell + u)$.

6. Experiments

6.1. Transductive Link Prediction

Once the problem of semi-supervised output kernel regression is solved, we come back to link prediction by building the following classifier as announced in Section 2: $\forall (o, o') \in \mathcal{O} \times \mathcal{O}$,

$$\hat{f}_\theta(o, o') = \mathrm{sgn}(\langle \hat{h}(o), \hat{h}(o') \rangle_{\mathcal{F}_y} - \theta)\ . \quad (10)$$

As the input kernel function κ_x is assumed to be known as well as the values on $\mathcal{O}_\ell \times \mathcal{O}_\ell$ of the output kernel function κ_y (that is, K_{Y_ℓ}), we can learn the classifier and make prediction on new data. Using the operator-valued kernel $\mathcal{K}_x$ defined by (5) and the corresponding solution $\hat{h}$ obtained in (9) gives: $\forall (o, o') \in \mathcal{O} \times \mathcal{O}$,

$$\begin{aligned}\langle \hat{h}(o), \hat{h}(o') \rangle_{\mathcal{F}_y} &= \left\langle C X_{\ell+u}^T x(o), C X_{\ell+u}^T x(o') \right\rangle_{\mathcal{F}_y} \\ &= x(o)^T X_{\ell+u} B^T K_{Y_\ell} B X_{\ell+u}^T x(o')\ ,\end{aligned}$$

with $B = U(K_{X_{\ell+u}} U^T U + \lambda_1 I_{\ell+u} + 2\lambda_2 K_{X_{\ell+u}} L)^{-1}$ and where $X_{\ell+u}$, $K_{X_{\ell+u}}$, U and $I_{\ell+u}$ are defined in Proposition 6. Varying the threshold θ in (10) allows us to build ROC and Precision-Recall curves.

In our experiments, we evaluate the approach in the transductive setting, assuming that all the nodes are known at the beginning of the learning phase and that only a subgraph defined on a subset of nodes is given. Note that a comparison with frameworks such as the *link propagation* proposed in (Kashima et al., 2009) would not be appropriate since they deal with a slightly different assumption. Indeed, in the *link propagation* framework, arbitrary interactions may be considered labeled while the Penalized Output Kernel Regression (POKR) framework requires a subgraph of known interactions.

6.2. Experimental protocol

We perform experiments on three kinds of datasets: a collection of synthetic datasets, a co-authorship network and a Protein-Protein Interaction (PPI) network. For different values of ℓ, the number of labeled nodes, we have randomly picked 10 times a subsample of training examples and used the remaining as testing examples. Labeled interactions correspond to interactions between two nodes from the training set and the goal is to complete the interaction matrix. Note that a 10% selection of labeled nodes actually corresponds to only 1% of labeled interactions. Performance is evaluated by the areas under the ROC and the Precision-Recall curves (respectively denoted Auc-roc and Auc-pr) averaged over 10 random choices of training sets.

The input kernel $K_{X_{\ell+u}}$ is build through a gaussian kernel, whose hyperparameter σ is chosen to maximize an information criterion ($\sigma = 5.7$). The output kernel is a diffusion kernel of parameter β. Another diffusion kernel of parameter β_2 is also used for the smoothing penalty instead of the graph Laplacian: $\exp(-\beta_2 L) = \sum_{i=0}^{\infty} \frac{(-\beta_2 L)^i}{i!}$. We set $W = K_{X_{\ell+u}}$. The hyperparameters λ_1, λ_2, β, and β_2 are selected by a 5-fold cross-validation procedure on the training set to maximize the Auc-roc.

6.3. Synthetic networks

We illustrate our method on synthetic networks. In these experiments, we want to measure the improvement brought by the semi-supervised method in extreme cases (i.e. for low percentage of labeled nodes) when the input kernel is a very good approximation of the output kernel. We produce the data by sampling random graphs from a Erdős-Renyi law. The sampled graphs contain 700 nodes and their densities[4] have respectively been fixed to 0.01, 0.02 and 0.03. The input feature vectors have been obtained by applying Kernel PCA on the diffusion kernel associated with the graph, whose diffusion parameter is chosen to maximize an information criterion. Finally, we use the components

[4]The graph density corresponds to the probability of presence of edges in the graph.

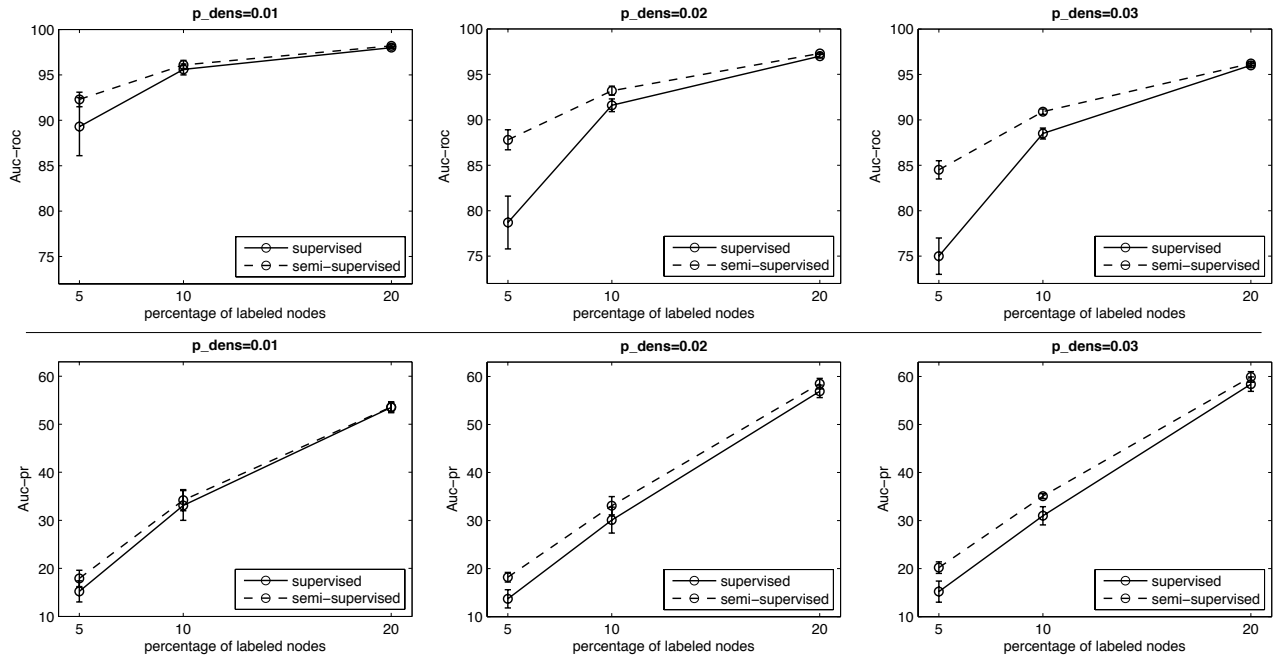

Figure 1. Averaged and standard deviation values of Auc-roc (top line) and Auc-pr (bottom line) for the reconstruction of three synthetics networks, given a percentage of 5%, 10% and 20% labeled nodes. The columns correspond to different graph densities (denoted p_dens) which are 0.01, 0.02 and 0.03 respectively.

that capture 95% of the variance as input features.

Figure 1 reports the averaged AUC and the standard deviations obtained for different densities of networks and for different percentage values of labeled nodes. One can observe that the semi-supervised approach improves upon the supervised one on both kinds of AUC, especially for a small percentage of labeled data (up to 10%). Based on these results one can formulate the hypothesis that supervised link prediction is harder in the case of more dense networks and that the contribution of unlabeled data seems more helpful in this case. One can also assume that using unlabeled data increases the AUCs for low percentage of labeled data. But when enough information can be found in the labeled data, semi-supervised learning does not improve the performance.

6.4. NIPS co-authorship network

We apply our method on a co-authorship dataset (Globerson et al., 2007) containing information on publications of the NIPS conferences from 1988 to 2003. The vertices of the network represent authors and an edge connects two authors if they have at least one joint publication. Among the 2865 authors, we consider only the ones with at least two links in the co-authorship network and we therefore focus on a network containing 2026 authors with an empirical link density of 0.002. Each author is described by a vector of 14036 values, corresponding to the frequency with which he uses each given word in his papers.

Averaged AUC results in both settings are shown in Table 1. As previously we can observe that the semi-supervised method improves the performance compared to the supervised method. For a percentage value of labeled authors of 5 %, this improvement reaches 4.8 in Auc-roc and 0.6 in Auc-pr.

6.5. Protein-protein interaction network

We also perform experiments on a PPI network of the yeast *Saccharomyces Cerevisiae* composed of 984 proteins linked by 2438 interactions. Several input features have already been used to infer this network: gene expressions, phylogenetic profiles, localization data and protein interaction data derived from yeast two-hybrid (Yamanishi et al., 2004; Kato et al., 2005; Geurts et al., 2006; Bleakley et al., 2007). Among these data, gene expression appears to be the most important source of information for this task. Therefore, we use the gene expressions as input features.

Supervised setting. Several supervised methods for network inference use this biological network as a

Table 1. Reconstruction of the NIPS co-authorship network for the authors with a minimum of two links in the network. The percentage values correspond to the proportions of labeled authors. The averaged Auc-roc and the Auc-pr are reported for the POKR method, together with the standard deviations, in the supervised and the transductive settings.

Methods	**Auc-roc**			**Auc-pr**		
	5%	**10%**	**20%**	**5%**	**10%**	**20%**
Supervised	71.1 ± 1.1	77.5 ± 0.7	82.2 ± 0.8	7.7 ± 1.8	15.9 ± 1.2	29.3 ± 2.7
Semi-supervised	75.9 ± 1.3	80.7 ± 0.8	84.7 ± 0.3	8.3 ± 1.3	17.2 ± 1.2	29.0 ± 0.9

benchmark. We complete the comparison provided in (Bleakley et al., 2007) with our method and the Output Kernel Tree with extra-trees method ($OK3 + ET$) (Geurts et al., 2007a). The protocol described in (Bleakley et al., 2007) is used: each method is evaluated through a 5-fold cross-validation experiment and the hyperparameters are tuned using the training folds. Auc-roc and Auc-pr are computed only for the possible interactions between proteins in the test set and proteins in the training set.

Table 2. Auc estimated by 5-CV for the yeast PPI network reconstruction from expression data in the supervised setting. The first three lines come from (Bleakley et al., 2007): *em* stands for em projection method (Tsuda et al., 2003), *Pkernel* for tensor product pairwise kernel with SVM (Ben-Hur & Noble, 2005) and *local* for local models with SVM (Bleakley et al., 2007). The results for $OK3 + ET$ (Geurts et al., 2007a) and POKR are also given.

Methods	**Auc-Roc**	**Auc-Pr**
em	80.6 ± 1.1	6.3 ± 1.2
Pkernel	83.8 ± 1.4	7.6 ± 1.0
local	78.1 ± 1.1	2.6 ± 0.4
OK3+ET	$\mathbf{84.6 \pm 1.4}$	11.2 ± 3.3
POKR	83.3 ± 2.1	$\mathbf{13.7 \pm 4.4}$

Table 2 reports the results presented in (Bleakley et al., 2007) that exhibit the best Auc-roc and Auc-pr, and the results for the OK3+ET and the POKR methods. In terms of Auc-roc, the POKR method behaves as well as the OK3+ET and the Pkernel methods, with a slight advantage for OK3+ET. Regarding Auc-pr, the POKR method achieves rather good performances compared to the others.

Transductive setting. Now, we can experiment the POKR method in the transductive setting following the experimental protocol described in 6.2.

Averaged and standard deviations of the Auc-roc and Auc-pr values are summarized in Table 3. It is worth noting that PPI network inference problems are characterized by a small number of labeled proteins, and we can observe that the semi-supervised method reaches slight improvement in this case.

7. Conclusion

We presented a new method for semi-supervised and transductive link prediction based on Output Kernel Regression. This recent framework allows to convert the problem of learning a pairwise classifier into the task of learning a single output kernel regressor, which means that functions of interest are vector-valued.

To achieve semi-supervised regression with a smoothing constraint that is known to be performing on semi-supervised setting, we started from the theory of RKHS with operator-valued kernels (Senkene & Tempel'man, 1973; Micchelli & Pontil, 2005). We stated and proved a new representer theorem devoted to semi-supervised learning in RKHS with vector-valued functions with a penalized least-square cost. Then, given a simple definition of the operator-valued kernel, we derived a close-formed solution that extends the reformulated KDE proposed by Cortes et al. (2005) to the semi-supervised case.

We extensively studied the behaviour of the provided models on transductive link prediction using artificial data and two real datasets: a protein-protein interaction network and a co-authorship network. The experiments show that using the unlabeled data improve performances for a very low percentage of known links.

Future works encompass the study of OKR with other choices of the operator-valued kernel as well as other data-dependent costs. Moreover, through this work we have shown a new application of RKHS theory with operator-valued kernel which is quite different from the existing ones. This theory opens a large avenue to the exploitation of complex and structured output data.

Acknowledgments

We are grateful to the reviewers for their valuable comments. This work was supported in part by University of Evry through a PhD grant and by French National Research Agency through the grant ANR09 SYSC009 01 ODESSA.

Table 3. Auc results for the reconstruction of the PPI network from gene expression data with the POKR method in the supervised and the semi-supervised settings. The percentage values correspond to the proportions of labeled proteins.

Methods	**Auc-roc**			**Auc-pr**		
	5%	**10**%	**20**%	**5**%	**10**%	**20**%
Supervised	76.9 ± 4.3	80.3 ± 0.9	82.1 ± 0.6	5.4 ± 1.6	7.1 ± 1.1	8.1 ± 0.7
Semi-supervised	79.6 ± 0.9	80.7 ± 1.0	81.9 ± 0.7	6.6 ± 1.1	7.6 ± 0.8	8.4 ± 0.5

References

Argyriou, A., Micchelli, C. A., and Pontil, M. When is there a representer theorem? Vector vs matrix regularizers. *J. of Machine Learning Res.*, 10, 2009.

Belkin, M. and Niyogi, P. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56, 2004.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Ben-Hur, A. and Noble, W. S. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(1):38–46, 2005.

Bleakley, K., Biau, G., and Vert, J.-P. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65, 2007.

Caponnetto, A., Micchelli, C. A., Pontil, M., and Ying, Y. Universal multitask kernels. *Journal of Machine Learning Research*, 9:1615–1646, 2008.

Cortes, C., Mohri, M., and Weston, J. A general regression technique for learning transductions. In *Proc. of the 22nd Intl. Conf. on Machine Learning*, pp. 153–160, 2005.

Geurts, P., Wehenkel, L., and d'Alché Buc F. Kernelizing the output of tree-based methods. In *Proc. of the 23th Intl. Conf. on Machine learning*, 2006.

Geurts, P., Touleimat, N., Dutreix, M., and d'Alché-Buc, F. Inferring biological networks with output kernel trees. *BMC Bioinformatics (PMSB06 special issue)*, 8(Suppl 2):S4, 2007a.

Geurts, P., Wehenkel, L., and d'Alché-Buc, F. Gradient boosting for kernelized output spaces. In *Proc. of the 24th Intl. Conf. on Machine Learning*, 2007b.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

Huynen, M. A., von Mering, C., and Bork, P. Function prediction and protein networks. *Current Opinion in Cell Biology*, 15(2):191–198, 2003.

Kadri, H., Duflos, E., Preux, P., Canu, S., and Davy, M. Nonlinear functional regression: a functional rkhs approach. In *JMLR Proc. of Intl. Conf. on Artificial Intelligence and Statistics*, volume 9, 2010.

Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., and Tsuda, K. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *Proc. of the 9th SIAM Intl. Conf. on Data Mining*, pp. 1099–1110, 2009.

Kato, T., Tsuda, K., and Asai, K. Selective integration of multiple biological data for supervised network inference. *Bioinformatics*, 21(10):2488–2495, 2005.

Kondor, R. I. and Lafferty, J. D. Diffusion kernels on graphs and other discrete input spaces. In *Proc. of the 19th Intl. Conf. on Machine Learning*, 2002.

Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *J. of the Am. Soc. for Information Science and Technology*, 58(7), 2007.

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

Miller, K., Griffiths, T., and Jordan, M. Nonparametric latent feature models for link prediction. In *Adv. in Neural Information Processing Systems 22*, 2009.

Senkene, E. and Tempel'man, A. Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4):665–670, 1973.

Taskar, B., Wong, M., Abbeel, P., and Koller, D. Link prediction in relational data. In *Advances in Neural Information Processing Systems 15*, 2003.

Tsuda, K., Akaho, S., and Asai, K. The em algorithm for kernel matrix completion with auxiliary data. *J. of Machine Learning Research*, 4:67–81, 2003.

Yamanishi, Y., Vert, J.-P., and Kanehisa, M. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20, 2004.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2004.