Regularized Output Kernel Regression applied to protein-protein interaction network inference

Céline Brouard
IBISC EA 4526, Université d'Évry
F-91025 Évry cedex, France
celine.brouard@ibisc.fr

Marie Szafranski
ENSIIE, IBISC EA 4526, Université d'Évry
F-91025 Évry cedex, France
marie.szafranski@ibisc.fr

Florence d'Alché-Buc
IBISC EA 4526, Université d'Évry
F-91025 Évry cedex, France
florence.dalche@ibisc.fr

1 Introduction

Link prediction in social networks as well as in biological networks [8, 12] have recently attracted considerable interest among machine learning community. In the area of social networks applications, many link predictors have been defined within the framework of probabilistic graphical models, building posterior probabilities [13, 15] while in the area of biological networks, a large number of methods have been built on kernel tools. In this paper, we address the link prediction issue within a recently introduced new learning framework called Output Kernel Regression and extend it to the case of a transductive setting. The approach was built to complete protein-protein interaction (PPI) networks but is in fact general and could be applied to other applicative domains.

In brief words, inference of protein-protein interaction networks is motivated by the cost and the difficulty to experimentally detect physical interactions between two proteins. It mainly relies on the idea that some known features of the proteins could help to suggest new physical interactions. In parallel to the community of link prediction in social networks, network inference approaches have been developed either based on a supervised framework or on matrix completion. Most of the supervised approaches aim at building a classifier whose input is a pair of proteins and output is a binary value that indicates the existence of an interaction between these proteins. In [2], a pairwise SVM based on tensor kernel is proposed to solve this task while [17] and [5, 7, 6] make predictions by thresholding a metric or a kernel that they learn from data. Finally, local approaches developed in [3] consider classifiers associated with one protein. From another point of view, PPI prediction can be seen as a matrix completion problem that fits into an unsupervised setting with some constraints [10, 16] or directly into a semi-supervised framework [9, 18].

In this work, we combine the advantages of both approaches by building a new family of input and output kernels-based regressors that can be used for supervised link prediction as well as for matrix completion. Instead of learning a pairwise classifier, we convert the supervised classification problem into a kernel learning problem and finally, address the task by learning a function whose output lies in an output feature space linked to the network at hand. In the rest of this paper, we first introduce the framework of Output Kernel Regression for link prediction. Then we propose a new family of models that can be used for supervised as well as semi-supervised learning as soon as it minimizes an appropriate defined penalized least square cost function. Solutions of the optimization problem are closed-form expressions when a ℓ_2-norm regularization is applied. Then we show experimental results obtained both in the supervised and the semi-supervised cases with an evaluation of the task as a transductive problem.

2 Link prediction with output kernel regression

Let us define $\mathcal{O}$ the set of objects we are interested in and $f : \mathcal{O} \times \mathcal{O} \to \{0, 1\}$, a classifier that predicts if two objects (individuals, proteins, documents) interact with each other. For a given learning set $\mathcal{O}_\ell = \{o_1, \cdots, o_\ell\}$, we assume that the following information is available:

- A set of input feature vectors $\mathbf{x}_1 = x(o_1), \cdots, \mathbf{x}_\ell = x(o_\ell)$ encoding some properties of the objects of $\mathcal{O}_\ell$.
- An output Gram matrix K_{Y_ℓ} that codes for the proximity of objects in terms of nodes in the known graph of interaction. The coefficients are supposed to be defined from a positive definite output kernel function $k_Y : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$ with $\forall i, j \leq \ell, K_{Y_\ell}(i, j) = k_Y(o_i, o_j)$. Given k_Y, there exists an Hilbert space $\mathcal{Y}$ and a feature map $y : \mathcal{O} \to \mathcal{Y}$, such that $\forall (o, o')$ in $\mathcal{O}$, we have $k_Y(o, o') = \langle y(o), y(o') \rangle_{\mathcal{Y}}$.

However, we should emphasize that we do not know the kernel function k_Y, we only know its values on $\mathcal{O}_\ell \times \mathcal{O}_\ell$. Typically, we use in this work the diffusion kernel matrix $K_{Y_\ell} = \exp(-\beta L_{Y_\ell})$ where $L_{Y_\ell} = D_\ell - W_\ell$, with W_ℓ the adjacency matrix given for the ℓ nodes and D_ℓ the corresponding degree matrix.

We propose to define a classifier by approximating the output kernel k_Y and thresholding it:

$$f_\theta(o, o') = \operatorname{sgn}(\hat{k}_Y(o, o') - \theta) . \tag{1}$$

To approximate k_Y, we build a function whose output is based on the dot product in $\mathcal{Y}$ of the images of o and o' by a single input function h:

$$f_\theta(o, o') = \operatorname{sgn}(\langle h(o), h(o') \rangle_{\mathcal{Y}} - \theta) . \tag{2}$$

Learning f reduces to learn h, a function whose output lies in a Hilbert space. Therefore, the idea is to use the kernel trick in the output space.

We will refer to this new learning task as Output Kernel Regression. It was first introduced in previous works [5, 7, 6] that presented the extension of tree-based methods to an output feature space.

In the following, we propose a new model family that uses the kernel trick both in the input and output spaces and enjoys closed-form solutions. We also extend the link prediction task to a semi-supervised setting.

3 Regularized input and output kernel regression

The training set is now symmetrically defined by an input Gram matrix K_{X_ℓ} as well as an output Gram matrix K_{Y_ℓ}. K_{X_ℓ} encodes for the properties of the objects of the training set $\mathcal{O}_\ell$. As in the output case, the coefficients of the Gram matrix are supposed to be defined from a positive definite input kernel function $k_X : \mathcal{O} \times \mathcal{O} \to \mathbb{R}$, with $\forall i, j \leq \ell, K_{X_\ell}(i, j) = k_X(o_i, o_j)$. Given k_X, there exists an Hilbert space $\mathcal{X}$ and a feature map $x : \mathcal{O} \to \mathcal{X}$, such that $\forall (o, o')$ in $\mathcal{O}$, we have $k_X(o, o') = \langle x(o), x(o') \rangle_{\mathcal{X}}$. Contrary to the output case, the function k_X is given.

Let us introduce models of the form: $h_M(o) = Mx(o) \in \mathcal{Y}$. If we take a model very close to SVM and Maximum Margin Robot [14], we get:

$$h_a(o) = \sum_{i=1}^{\ell} a_i y(o_i) k_X(o_i, o) = (Y_\ell I_a X_\ell^T) x(o) ,$$

where $I_a = \operatorname{diag}(a)$, $X_\ell = [x(o_1), \cdots, x(o_\ell)]$ is a matrix of dimension $\dim(\mathcal{X}) \times \ell$ and $Y_\ell = [y(o_1), \cdots, y(o_\ell)]$ a matrix of dimension $\dim(\mathcal{Y}) \times \ell$.[1]

If we extend this model using an arbitrary $\ell \times \ell$ matrix A_ℓ, we get the following definition:

$$h_{A_\ell}(o) = \left(Y_\ell A_\ell X_\ell^T\right) x(o) .$$

[1] In what follows, the same convention will apply to matrices X_n and Y_n.

Finally if as in [4], we take a general matrix, whose dimension is $\dim(\mathcal{Y}) \times \dim(\mathcal{X})$, we have a very general model:

$$h_A(o) = Ax(o)\,.$$

To estimate the parameters of these models, we can minimize a square loss function while controlling the complexity of the model h_M:

$$\min \sum_{i=1}^{\ell} \|h_M(o_i) - y(o_i)\|^2 + \lambda_1 \|M\|_F^2\,.$$

4 Semi-supervised regularized output kernel regression

Noticing that in some link prediction problem such as protein-protein network inference, we know the whole set of nodes (proteins) of interest and this set is finite ($|\mathcal{O}| = n$), we propose a transductive learning task. It consists in using the input information concerning the unlabeled data during the training phase. In that context, we assume that if we know K_{X_n} as input information, we only know a subgraph of interactions for a subset $\mathcal{O}_\ell$ of $\mathcal{O}_n$ with $\ell \ll n$. The goal is to complete the missing values of K_{Y_n}, the Gram matrix that should be defined for all the proteins of the whole training set. Although the task is transductive, we solve it using semi-supervised output kernel regression and build a regressor that will only be used to complete the data at hand. To take advantage of the known inputs of the unlabeled data, we consider cost functions that include a smoothness constraint on the regressor to build. Many works have emphasized the efficiency of such a constraint based on Laplacian operators [1, 11]. In this new setting, the cost function we consider for minimization is now:

$$\min \sum_{i=1}^{\ell} \|h_M(o_i) - y(o_i)\|^2 + \lambda_1 \|M\|_F^2 + \lambda_2 \operatorname{trace}(h_M L_{X_n} h_M^T)\,,$$

where L_{X_n} is the diffusion kernel associated to the graph Laplacian $D_n - K_{X_n}$, with $L_{X_n} = \exp(-\beta(D_n - K_{X_n}))$, D_n the diagonal degree matrix given by $d_{ii} = \sum_{j=1}^{n} k_X(o_i, o_j)$, and where λ_1 and λ_2 are positive regularization parameters. The third term codes for a smoothness constraint that can be applied to outputs of h for both labeled and unlabeled data.

4.1 Solutions in the supervised and semi-supervised cases

As in (kernel) ridge regression, minimizing each of the cost functions presented above leads to closed-form solutions that are briefly presented in table 1 for the supervised case ($\lambda_2 = 0$) and the semi-supervised case ($\lambda_2 > 0$).

h_a	Sup.	$\hat{a} = (K_{Y_\ell}.(K_{X_\ell}K_{X_\ell} + \lambda_1 K_{X_\ell}))^{-1} diag(K_{Y_\ell}K_{X_\ell})$
	Semi-sup.	$\hat{a} = (K_{Y_\ell}.(K_{X_\ell}K_{X_\ell} + \lambda_1 K_{X_\ell} + 2\lambda_2 K_{X_{\ell n}} L_{X_n} K_{X_{n\ell}}))^{-1} diag(K_{Y_\ell}K_{X_\ell})$
h_{A_ℓ}	Sup.	$\hat{A}_\ell = (K_{X_\ell} + \lambda_1 I_\ell)^{-1}$
	Semi-sup.	$\hat{A}_\ell = K_{X_\ell}(K_{X_\ell}K_{X_\ell} + \lambda_1 K_{X_\ell} + 2\lambda_2 K_{X_{\ell n}} L_{X_n} K_{X_{n\ell}})^{-1}$
h_A	Sup. [4]	$\hat{A} = Y_\ell(K_{X_\ell} + \lambda_1 I_\ell)^{-1} X_\ell^T$
	Semi-sup.	$\hat{A} = Y_\ell V_\ell (K_{X_n} V_\ell^T V_\ell + \lambda_1 I_n + 2\lambda_2 K_{X_n} L_{X_n}))^{-1} X_n^T$

Table 1: Solutions for the models h_a, h_{A_ℓ} and h_A in the supervised and semi-supervised settings. $M.M'$ denotes the element-wise product between matrices M and M'; I_n and I_ℓ are respectively identity matrices of size n and ℓ; V_ℓ denotes a $\ell \times n$ matrix that contains an identity matrix of size ℓ on the left hand side and a zero matrix of size $\ell \times (n - \ell)$ on the right hand side.

5 Experiments

We illustrate our method on a PPI network of the yeast *Saccharomyces Cerevisiae* composed of 984 proteins linked by 2438 interactions. To reconstruct the PPI network, we deal with usual input features that are gene expression data, phylogenetic profiles, protein localization and protein interaction data derived from yeast two-hybrid (see for instance [3, 5, 7, 10, 17] and references therein for a more complete description of data). Gene expression data correspond to time series and are handled through a RBF kernel matrix. Phylogenetic profiles are binary vectors of size 145, each value coding for the presence or the absence of an orthologous protein in a given organism. Similarly, localization data give vectors of 23 binary values coding for the presence or absence of the protein in a given intracellular location. The linear kernel is used for both datasets. The yeast two-hybrid network is transformed into a diffusion kernel of parameter $\beta = 1$. We also use an integrated kernel which is the sum of the four kernels.

Supervised setting. We compared our method in the supervised setting following the protocol used in [3]. We first evaluated the method through a 5-fold cross-validation procedure, and tuned the hyperparameters using the training folds. Table 2 reports the tests in [3] that exhibit the best areas under the ROC and the FDR curves (respectively denoted AUC and AUF). We added the results for the Regularized Output Kernel Regression (ROKR) method. Table 2 reports the results for the models of ROKR that exhibit the best AUC and AUF values : A and A_ℓ (which are the same in the supervised setting).[2] In terms of AUC, the ROKR gives the best result using the integrated dataset while it does not perform as well as the others using only the protein localization, the phylogenetic profiles or the protein yeast two-hybrid data. It achieves an equivalent result compared with the Pkernel method for the gene expression data. Regarding the AUF, the ROKR obtains similar results than the others using the protein localization and the phylogenetic profiles data but achieves quite good performances for the gene expression, the protein yeast two-hybrid and the integrated datasets.

	Methods	exp	loc	phy	y2h	int
AUC	em	80.6 ± 1.1	76.7 ± 3.8	71.0 ± 1.3	57.2 ± 2.7	89.3 ± 1.1
	Pkernel	$\mathbf{83.8 \pm 1.4}$	$\mathbf{79.2 \pm 2.6}$	74.8 ± 1.8	67.5 ± 1.8	87.2 ± 0.8
	local	78.1 ± 1.1	77.1 ± 2.9	$\mathbf{75.5 \pm 2.4}$	$\mathbf{77.8 \pm 1.2}$	87.6 ± 1.8
	ROKR (A/A_ℓ)	83.3 ± 2.1	69.2 ± 1.8	69.6 ± 1.5	60.8 ± 3.5	$\mathbf{91.0 \pm 0.4}$
AUF	em	93.7 ± 1.2	$\mathbf{94.5 \pm 1.1}$	$\mathbf{96.8 \pm 0.5}$	89.6 ± 1.0	80.9 ± 1.3
	Pkernel	92.4 ± 1.0	95.1 ± 1.0	98.2 ± 0.3	98.5 ± 0.5	88.6 ± 2.2
	local	97.4 ± 0.4	96.3 ± 0.9	97.9 ± 0.3	92.4 ± 1.6	74.5 ± 3.4
	ROKR (A/A_ℓ)	$\mathbf{86.3 \pm 4.4}$	95.2 ± 0.8	97.4 ± 0.4	$\mathbf{87.1 \pm 2.9}$	$\mathbf{72.8 \pm 6.5}$

Table 2: Reconstruction of the PPI network from gene expression data (exp), protein localization (loc), phylogenetic profiles (phy), protein interaction data derived from yeast two-hybrid (y2h) and the integration of the individual datasets (int). The AUC and the AUF are estimated with a 5-fold cross-validation procedure. The first three lines have been obtained and presented in [3]. *em* stands for em projection method; *Pkernel* for tensor product pairwise kernel with SVM; *local* for local models with SVM; and *ROKR (A/A_ℓ)* stands for the A or A_ℓ models (which are the same in the supervised setting).

Semi-supervised setting. We then experimented the ROKR method in the semi-supervised setting and compared the results with those obtained in the supervised setting for the different models. For different values of ℓ, that is the number of labeled proteins, we randomly sub-sampled a training set of proteins and considered all the remaining proteins for the test set. The interactions assumed to be known are those between two proteins from the training set. *Therefore, a percentage value of labeled proteins of 10% actually corresponds to a percentage of labeled interactions of 1%.* We ran each experiment ten times and tuned the hyperparameters similarly on the training set, using only expression data as input feature. Table 3 summarizes the averaged values of AUC and AUF. The models A and A_ℓ exhibit better AUC and AUF values than the model a. One can also observe that the semi-supervised methods reach better performances when the number of labeled proteins is small, which is usually the case in protein-protein interaction network inference problems. Although the models A and A_ℓ share similar results, it is worth pointing out that the model A_ℓ has a lower

[2] Note that a comparison with frameworks such as the *link propagation* proposed in [9] 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 ROKR framework requires a subgraph of known interactions.

computational complexity than the model A since it requires the inversion of a matrix of size $\ell \times \ell$ while the model A needs the inversion of a matrix of size $n \times n$.

Methods	AUC			AUF		
	$\mathbf{10\%}$	$\mathbf{20\%}$	$\mathbf{50\%}$	$\mathbf{10\%}$	$\mathbf{20\%}$	$\mathbf{50\%}$
Sup. (a)	66.6 ± 7.4	72.2 ± 3.0	75.1 ± 1.3	96.8 ± 2.1	94.8 ± 1.3	94.3 ± 0.5
Semi-sup. (a)	73.7 ± 2.9	76.6 ± 1.7	78.9 ± 0.9	95.4 ± 1.2	94.2 ± 0.8	93.9 ± 0.5
Sup. (A/A_p)	74.6 ± 3.2	81.3 ± 2.0	83.7 ± 0.7	95.7 ± 1.6	91.9 ± 1.5	90.9 ± 1.3
Semi-sup. (A_p)	78.9 ± 3.1	81.8 ± 1.0	84.3 ± 0.5	93.7 ± 1.7	92.2 ± 1.3	90.9 ± 1.4
Semi-sup. (A)	79.1 ± 2.7	82.1 ± 0.9	84.3 ± 0.5	93.7 ± 1.5	92.2 ± 1.3	91.3 ± 1.4

Table 3: Reconstruction of the protein-protein interaction network from the gene expression data. The percentage values correspond to the proportions of labeled proteins. The AUC and the AUF are reported for the ROKR method, in the supervised and the semi-supervised settings.

6 Conclusion

We presented a new method for 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. A new family of models, with both input and output kernels, are learnt by minimizing least square loss penalized by a ℓ_2 norm. Both in the supervised and the semi-supervised case, we obtain closed-form solutions as for "classic" kernel ridge regression. Experimental results on protein-protein interaction networks show that the new methods exhibit better performances. Future work encompasses further comparison with other link prediction methods, application to other fields and input kernel selection.

References

[1] M. Belkin and P. Niyogi. Towards a theoritical foundation for laplacian-based manifold methods. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 486–500, 2005.

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

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

[4] 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.

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

[6] P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Gradient boosting for kernelized output spaces. In *ACM International Conference Proceeding Series (Proceedings of the 24th International Conference on Machine Learning)*, volume 227, pages 289–296. ACM, 2007.

[7] P. Geurts, L. Wehenkel, and d'Alché Buc F. Kernelizing the output of tree-based methods. In *Proceedings of the 23th international conference on Machine learning*, pages 345–352, 2006.

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

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

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

[11] J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, pages 801–808. MA, MIT Press, 2008.

[12] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007.

[13] K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems 22*, pages 1276–1284, 2009.

[14] S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, UK, 2005.

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

[16] K. Tsuda and W. S. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20(1):326–333, 2004.

[17] Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20:i363–i370, 2004.

[18] K. Y. Yip and M. Gerstein. Training set expansion: an approach to improving the reconstruction of biological networks from limited and uneven reliable interactions. *Bioinformatics*, 25(2):243–250, 2009.