Link transfer for improving protein-protein interaction prediction using multiple species

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

Protein-protein interaction (PPI) network inference has attracted interest of machine learning researchers as a typical problem of structured data mining. Like link prediction in social networks, PPI prediction can be solved using supervised network inference approaches if one considers to build a classifier whose input is a pair of nodes and output, a binary value that codes for the presence of a physical interaction between two proteins. The training data used for this task are usually input feature vectors that represent information about the proteins and a given adjacency matrix that codes for the known interactions. Among supervised link prediction approaches, let us cite pairwise SVM based on tensor kernel [3], metric or kernel learning [12] and [7, 8, 6] and local approaches developed in [4]. In parallel, bioinformatics researchers have defined other strategies that consist, for example, in mapping known interactions between a reference organism onto a target organism and this for the orthologous genes: this is called the protein-protein interologs approach [10]. As far as PPI networks as well as the homology between protein sequences are available for potential reference organisms, this strategy sounds relevant if data are not too noisy. In this work, we define a new task of link prediction, we call it "link transfer", that resembles the interolog approach while remaining in the supervised learning framework. The underlying idea of link transfer is to use PPI networks of other species to constrain the training of a supervised predictor of PPI in a target species. Contrary to Kashima et al. [9], we do not assume that there exists input information for the additional species but only output information. This paradigm thus differs from transfer learning or multi-task learning [5, 2] but corresponds to a realistic setting of PPI network inference.

Building up upon previous works on Output Kernel Regression [7, 8, 6] where an output kernel is learned to build the classifier, we formulate the new task in the framework of output kernel learning and investigate how to incorporate the information available from the reference species in order to improve the performance of the output kernel regressor. We propose to use output kernel regression twice, first to convert output feature vectors from a reference species to the target species and then to learn the target network. The underlying idea of the converter is to increase the training set of the target species by converting the output space of the reference species to the output space of the target species. In Section 2 we describe the general framework of output kernel regression for PPI network inference and its extension to link transfer. In Section 3 we evaluate it empirically using yeast as the target species.

2 Regularized Output Kernel Regression

Let us introduce the general framework of Output Kernel Regression for protein-protein network inference. We consider a single target species. Let $\mathcal{O}$ be the set of proteins in the target species. During the training phase, $\mathcal{O}_n$ a subset of n proteins, and W_n the adjacency matrix given for the interactions between the corresponding n proteins are given. These available data are encoded into:

- An input Gram matrix K_{X_n} whose coefficients are supposed to be defined from some positive definite kernel function: $\forall\, i \leq n, j \leq n,\ K_{X_n}(i,j) = \kappa_X(o_i, o_j)$.
- Another Gram matrix K_{Y_n} that codes for the proximity of proteins as nodes in the interaction graph only known for the proteins of $\mathcal{V}_n$. We use here the diffusion kernel matrix $K_{Y_n} = \exp(-\beta L_{Y_n})$ where $L_{Y_n} = D_n - W_n$ with W_n the adjacency matrix given for the n proteins and D_n the corresponding degree matrix.

Let us imagine that we know $\kappa_Y : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{R}$, the positive definite kernel that encodes the proximity of proteins in terms of nodes in the interaction graph of a target species, $\mathcal{Y}$ the associated feature space endowed with kernel κ_Y as a dot product and $y(\cdot) : \mathcal{O} \rightarrow \mathcal{Y}$, the feature map such that $\forall\, o, o',\ \kappa_Y(o, o') = \langle y(o), y(o') \rangle$ and especially: $\forall\, o_i, o_j \in \mathcal{O}_n, \kappa_Y(o_i, o_j) = K_{Y_n}(i,j)$. Let us call $f : \mathcal{O} \times \mathcal{O} \rightarrow \{0, 1\}$ a classifier whose input is a pair of proteins features and outputs a binary value that indicates if there is a interaction or not between those proteins. Knowing κ_Y we can define the classifier f by thresholding the kernel : $f_\theta(o, o') = \mathrm{sgn}(\kappa_y(o, o') - \theta)$. However, we do not know κ_Y but only the corresponding Gram matrix K_{Y_n}, defined for the proteins of the training set. In the framework of output kernel regression, we propose to approximate κ_Y by using a dot product between images of the single input function $h : \mathcal{O} \rightarrow \mathcal{Y}$, $f_\theta(o, o') = \mathrm{sgn}(\langle h(o), h(o') \rangle - \theta)$. Learning f reduces to learn h, a function that uses the kernel trick in the output space. This new learning task has been referred as Output Kernel Regression in previous works [7, 8, 6] and was tackled by extending regression trees to output kernel feature space. In this work we focus on Regularized Output Kernel Regression (ROKR), a recently proposed model [1] that shares the same form as SVMs and Maximum Margin Robot [11]:

$$h_a(o) = \sum_{i=1}^{n} a_i y(o_i) \kappa_X(o_i, o)\ . \tag{1}$$

The model h_a can be learned by minimizing a regularized least square loss:

$$\min \sum_{i=1}^{n} \parallel h_a(o_i) - y(o_i) \parallel^2 + \lambda_1 \parallel a \parallel^2\ , \tag{2}$$

for which a closed-form solution exists: $\hat{a} = (K_{Y_n} * (K_{X_n} K_{X_n} + \lambda_1 K_{X_n}))^{-1} \mathrm{diag}(K_{Y_n} K_{X_n})$.Thus we obtain in this case the following approximation for κ_Y:

$$\hat{\kappa}_Y(o, o') = \sum_{i,j} \hat{a}_i \hat{a}_j \kappa_Y(o_i, o_j) \kappa_X(o_i, o) \kappa_X(o_j, o')\ . \tag{3}$$

Link Transfer with ROKR

Let us now consider an additional species, call it species 1, for which we know the adjacency matrix W_1 that represents the physical interactions a set of proteins. For this reference species, we are missing the associated input features of the proteins. However, we have the list of proteins (genes) of the target species that have orthologs in the species 1. For sake of simplicity, we will use the same notations for a protein of the target species and its corresponding ortholog in the reference species. The link transfer task consists in adding the information contained in the PPI network of species 1 to help the prediction task for the target species. We notice that the two adjacency matrices W of the target species and W_1 of the reference species define two different Hilbert spaces: the Hilbert space H spanned by the images of $y(o_i), i = 1...n$ and the Hilbert space H_1 spanned by the images of $y_1(o_i), i = 1...p$. In order to cope with these two different spaces, we use an output kernel regressor $h_{1 \rightarrow t}$ that converts for a given protein o, $y_1(o)$ into $y(o)$.

The connection between the target and the reference species is a set of ortholog proteins, i.e., a subset of $\mathcal{O}$ has a one-to-one correspondence with a subset of $\mathcal{O}_1$. Let $\mathcal{O} = \{o_1, \ldots o_p\} \cup \{o_{p+1}, \ldots o_n\}$ and $\mathcal{O}_1 = \{o_1^1, \ldots o_p^1\} \cup \{o_{p+1}^1, \ldots o_{n_1}^1\}$ then

$$o_1 \longleftrightarrow o_1^1\ , \ldots , o_p \longleftrightarrow o_p^1\ .$$

The transfer learning is based on a converter function from the reference species to the target species. The idea is to increase the training information on which the mapping h is learned by incorporating

the data from the reference species. Let O_{train} be the set of orthologs whose absence/presence of links in the target species is known (orthologs in the train set) and $O_{transfer}$ be the set of orthologs whose absence/presence of links in the target species is not known (orthologs in the transfer set). The mapping h is learned by solving the following optimization problem which leads to a closed form solution as the problem in Equation (2):

$$\begin{aligned}\underset{a}{\operatorname{argmin}} \sum_i ||y(o_i) - h(o_i)||^2_{\mathcal{Y}} + \lambda ||a||^2 \\ + \lambda_{transfer} \sum_{i \in O_{transfer}} ||h_{1\to t}(y_1(o_i)) - h(o_i)||^2_{\mathcal{Y}} \,, \end{aligned} \tag{4}$$

with the last term transferring the information from the reference to the target species and $\lambda_{transfer} \geq 0$. The converter $h_{1\to t}$ maps the output space of the reference species ($\mathcal{Y}_1$) to the output space of the target species ($\mathcal{Y}$). This converter function is learned on the set of orthologs whose links are known both in the target and in the reference species, i.e., the orthologs from the training set:

$$\begin{aligned}\underset{h_{1\to t}}{\operatorname{argmin}} \sum_{i \in O_{train}} ||y(o_i) - h_{1\to t}(y_1(o_i))||^2_{\mathcal{Y}} + \lambda_{converter} ||b||^2 \,, \\ h_{1\to t}(y_1(o)) = \sum_{j \in O_{train}} b_j y(o_j) \left\langle y_1(o), y_1(o_j) \right\rangle_{\mathcal{Y}_1} \,. \end{aligned} \tag{5}$$

This idea can be extended to include the information from multiple reference species by adding extra terms in the optimization from Equation (4), each extra term corresponding to one reference species.

3 Empirical Evaluation

In this section we evaluate empirically the transfer learning approach for PPI prediction.

Data. We considered the baker's yeast (*Saccharomyces cerevisiae*) as the target organism. We used the yeast PPI network data of high-confidence physical protein-protein interactions also used in [12, 4, 6]. It consists of 2438 interactions that link 984 proteins. Each protein was associated with its gene expression, its location and its phylogenetic profile which was used to construct the input kernel. The following species were considered as reference species: *Schizosaccharomyces pombe* –fission yeast, *Mus musculus* –house mouse, *Arabidopsis thaliana* –plant. The PPI networks of the reference species were extracted from the String.db database (*http://string-db.org/*). This database has 7 types of interactions between proteins (neighborhood, fusion, occurrence, coexpression, experiments, database, textmining) from which we considered only the interactions which come from experiments. The set of orthologs between the target species and each of the reference species was obtained from the Inparanoid database (*http://inparanoid.sbc.su.se/*). The fission yeast has 271 orthologs with the target species, the mouse has 147 orthologs and the plant has 120 orthologs.

Protocol. We conducted experiments on the data set described above to determine whether the extra term (or terms for multiple reference species) in the optimization from Equation (4) improves the performance. The performance was evaluated as a function of the parameter $\lambda_{transfer}$. We fixed the other parameters of the model except $\lambda_{transfer}$ to its optimal values determined in the no-transfer case, i.e., $\sigma = 4$, $\beta = 3$ and $\lambda = 0.0001$ and we also fixed $\lambda_{converter} = 0.0001$. Further, the data set was randomly split 10 times into training and testing with different percentage for the size of the training data 10%, 15% and 20%. The model was learned on the training set for $\lambda_{transfer} \in 0 : 0.1 : 1$ and the performance was measured using area under the ROC curve (AUC) computed on the testing set.

Results. Figure 1 plots the AUC values as a function of the parameter $\lambda_{transfer}$. The three plots on the left side correspond to three sizes of the training data, 10%, 15% and 20% and one reference species, the fission yeast. The error bars give the standard deviation to the mean for the 10 runs. The optimal value $\lambda_{transfer} > 0$ suggests that the information from the reference species improves the performance. The improvement is bigger for a small size of the training set and decreases as the training set gets bigger, which is a behavior observed in most of the multi-task learning situations. The plots on the right-hand side are an extension of the three plots from the left-hand side to multiple reference species: results for one reference species (fission yeast) are plotted with solid lines, results for two reference species (fission yeast and plant) are plotted with dashed lines, and results for three

reference species (fission yeast, plant and house mouse) are plotted with dotted lines. The plots suggests that including multiple reference species as multiple sources of information increases the performance.

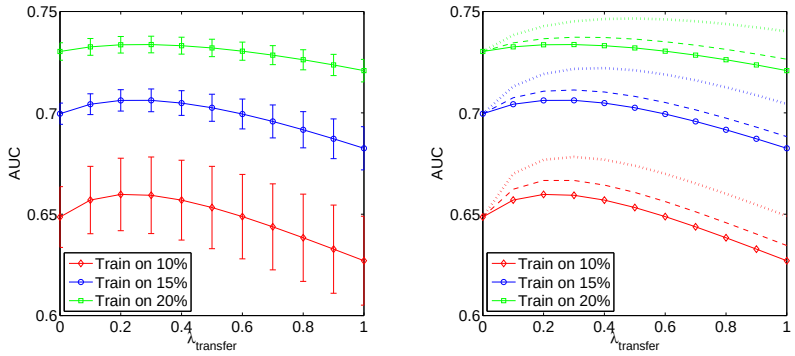

Figure 1: Plots of the AUC values as a function of the parameter $\lambda_{transfer}$. Left: The three plots correspond to three sizes of the training data, 10%, 15% and 20%, the error bars give the standard deviation to the mean for the 10 runs. Right: The plots are an extension of the three plots from the left-hand side to multiple reference species: the solid lines are the results obtained one reference species (fission yeast), the dashed lines are the results obtained with two reference species (fission yeast and plant), and the dotted lines are the results for three reference species (fission yeast, plant and house mouse).

References

[1] anonymous. Semi-supervised protein-protein interaction network inference with regularized output kernel regression. In *MLCB*, 2010.

[2] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

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

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

[5] T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.

[6] P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc. Inferring biological networks with output kernel trees. *BMC Bioinformatics*, 2007.

[7] P. Geurts, L. Wehenkel, and F. d'Alché Buc. Kernelizing the output of tree-based methods. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 345–352, New York, NY, USA, 2006. ACM.

[8] 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.

[9] H. Kashima, Y. Yamanishi, Ts. Kato, M. Sugiyama, and K. Tsuda. Simultaneous inference of biological networks of multiple species from genome-wide data and evolutionary information. *Bioinformatics*, 25(22):2962–2968, 2009.

[10] Magali Michaut, Samuel Kerrien, Luisa Montecchi-Palazzi, Franck Chauvat, Corinne Cassier-Chauvat, Jean-Christophe Aude, Pierre Legrain, and Henning Hermjakob. Interoporc: automated inference of highly conserved protein interaction networks. *Bioinformatics*, 24(14):1625–1631, 2008.

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

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