Regularized Output Kernel Regression for Protein-Protein Interaction Prediction: Application to Link Transfer and Transduction

Florence d'Alché-Buc[2], Adriana Bîrluţiu[1], Celine Brouard[2], Tom Heskes[1], Marie Szafranski[1]

[1] Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
[2] Université d'Evry-Val d'Essonne, France

Abstract. This study investigates transfer learning applied to output kernel regression for protein-protein interactions network inference.

1 Introduction

There has been a recent interest in devising new techniques and improving existing ones for mining various structured data types such as the ones arising, for example, in biology. Mining the protein interactions graph of a species, also known as protein-protein interactions (PPI) network inference [9], can give useful information to biologists about which proteins might interact.

A technique that has been recently applied to protein-protein network inference is output kernel regression [10, 2]. The idea of this method is to learn a mapping from inputs to a feature space associated with the outputs. The key aspect of this method is that the existing structure in the outputs can be exploited in the learning.

We consider a target species for which we want to predict the unknown links in its protein interactions graph. With each node in the graph there is associated information about that protein, such as gene expression data, location information or phylogenetic profile. Furthermore, the PPI networks from other species and the set of orthologs between the proteins in the target species and each of the reference species are also available. This data available from other species can be used to improve the performance of predictions of unknown links in the target species. This paradigm, known as transfer learning or multi-task learning, has been applied in various learning situations [4, 1]. In this study, we investigate how to incorporate the information available from the reference species, i.e., transfer learning, in order to improve the performance of the output kernel regression for protein-protein network inference. We propose a method based on a so-called converter function from the reference species to the target species. 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 for transfer learning. In Section 3 we evaluate it empirically using yeast as the target species. In Section 4 we conclude and give some directions for future research.

2 Framework

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the undirected graph modeling the PPI network of the target species, where $\mathcal{V}$ is the set of vertices and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges. We assume that each vertex v can be described by some features in some input space $\boldsymbol{x} \in \mathcal{X}$.

The input data consists of the information associated with each vertex of the graph and the output data is the adjacency matrix of the graph. The key aspect of this learning problem is the structure in the outputs that can be exploited in the learning. In order to do this, we construct a positive definite symmetric kernel $\kappa^{\psi} : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ derived from the adjacency matrix of the graph. This kernel implicitly defines a feature map $\psi : \mathcal{V} \to \mathcal{H}_{\psi}$ such that

$$\kappa^{\psi}(v, v') = \langle \psi(v), \psi(v') \rangle_{\mathcal{H}_{\psi}} \ .$$

We choose for κ^{ψ} the diffusion kernel [7], which has the following matrix associated with it

$$\boldsymbol{K}^{\psi} = \exp(-\beta \boldsymbol{L}), \quad \text{where} \quad \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \,,$$

with $\boldsymbol{D}$ the diagonal matrix of node connectivities, $\boldsymbol{A}$ the adjacency matrix, and $\beta > 0$ a parameter which controls the degree of diffusion. $\kappa^{\psi}(v, v')$ is high for adjacent vertices and small for non-adjacent ones, thus, $\psi(v)$ is close to $\psi(v')$ in $\mathcal{H}_{\psi}$ when there is an edge between the vertices v and v'. The diffusion kernel is a way to capture the long-range relationships between data points induced by the local structure of the graph.

In the same way, a positive definite symmetric kernel $\kappa^{\phi} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ constructed on the inputs implicitly defines a feature map $\phi : \mathcal{X} \to \mathcal{H}_{\phi}$ such that

$$\kappa^{\phi}(\boldsymbol{x}, \boldsymbol{x'}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x'}) \rangle_{\mathcal{H}_{\phi}} \ .$$

This kernel can be computed using, for example, the Gaussian kernel

$$\kappa^{\phi}(\boldsymbol{x}, \boldsymbol{x'}) = \kappa_{\text{Gauss}}(\boldsymbol{x}, \boldsymbol{x'}) \equiv \exp\left(-\frac{1}{2\sigma^2} \sum_j (x_j - x'_j)^2 \right) \ .$$

The idea of output kernel regression is to learn a mapping from inputs to the feature space associated with the outputs. We choose this mapping as

$$h : \mathcal{X} \to \mathcal{H}_{\psi}, \ h(x) = \sum_i \alpha_i \psi(v_i) \, \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle_{\mathcal{H}_{\phi}} \ . \tag{1}$$

where i runs over the training points. This choice for h has the advantage that even though it is define using the feature map ψ, when computing inner products it uses only kernel values and does not need to access the implicit feature map ψ, i.e.,

$$\langle h(\boldsymbol{x}), h(\boldsymbol{x}')\rangle = \sum_{i,j} \alpha_i \alpha_j \langle \psi(v_i), \psi(v_j)\rangle_{\mathcal{H}_\phi} \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}_\phi} \ .$$

The mapping h is an approximation to ψ, thus, in order to determine if there is an edge between the vertices v and v' we will threshold the value of the output kernel

$$\kappa^{\psi}(v, v') = \langle h(\boldsymbol{x}), h(\boldsymbol{x}')\rangle \ .$$

The mapping h can be learned by solving the following optimization problem

$$\underset{h}{\operatorname{argmin}} \sum_i ||\psi(v_i) - h(\boldsymbol{x}_i)||^2_{\mathcal{H}_\psi} + \lambda ||h||^2 \ . \tag{2}$$

where $\lambda > 0$ is a regularizer. There is a closed-form solution to the optimization problem from above.

Transfer Learning

Let $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ be the undirected graph modeling the protein interactions graph of a reference species. For the reference species we know only the adjacency matrix of the graph and do not have information associated with the vertices like in the case of the target species. As we did for the target species, we consider the implicit feature map for the PPI network of the reference species, ψ_1, and the space associated with it, $\mathcal{H}_{\psi_1}$. The connection between the target and the reference species is a set of ortholog proteins, i.e., a subset of $\mathcal{V}$ has a one-to-one correspondence with a subset of $\mathcal{V}_1$. Let $\mathcal{V} = \{v_1, \dots v_o\} \cup \{v_{o+1}, \dots v_n\}$ and $\mathcal{V}_1 = \{v_1^1, \dots v_o^1\} \cup \{v_{o+1}^1, \dots v_{n_1}^1\}$ then

$$\begin{aligned} v_1 &\longleftrightarrow v_1^1 \ , \\ &\dots \\ v_o &\longleftrightarrow v_o^1 \ . \end{aligned}$$

The transfer learning is based on a converter function from the reference species to the target species. The idea is to increase the training set for the target species 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_{test} be the set of orthologs whose absence/presence of links in the target species is not known (orthologs in the test set). The mapping h is learned by solving the following optimization problem:

$$\begin{aligned} \underset{h}{\operatorname{argmin}} \sum_i ||\psi(v_i) - h(\boldsymbol{x}_i)||^2_{\mathcal{H}_\psi} + \lambda ||h||^2 \\ + \lambda_{transfer} \sum_{i \in O_{test}} ||g_{1\to t}(\psi_1(v_i)) - h(\boldsymbol{x}_i)||^2_{\mathcal{H}_\psi} \end{aligned} \tag{3}$$

with the last term transferring the information from the reference to the target species and $\lambda_{transfer} \geq 0$. The converter $g_{1\to t}$ maps the output space of the reference species ($\mathcal{H}_{\psi_1}$) to the output space of the target species ($\mathcal{H}_{\psi}$). 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:

$$\underset{g_{1\to t}}{\operatorname{argmin}} \sum_{i\in O_{train}} ||\psi(v_i) - g_{1\to t}(\psi_1(v_i))||^2_{\mathcal{H}_{\psi_t}} + \lambda_{converter}||g_{1\to t}||^2 ,$$

$$g_{1\to t}(\psi_1(v)) = \sum_{j\in O_{train}} \beta_j \psi(v_j) \langle \psi_1(v), \psi_1(v_j)\rangle_{\mathcal{H}_{\psi_1}} . \quad (4)$$

The definition of the converter function from above has the same form and advantages as the definition of the prediction map from Equation (1).

This idea can be extended to include the information from multiple reference species by adding extra terms in the optimization from Equation (3), each extra term corresponding to one reference species.

3 Empirical Evaluation

In this section we evaluate empirically the transfer learning approach for PPI network inference described in the previous section.

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 [5]. It consists of 2438 interactions that link 984 proteins. Each protein has associated with it information about its gene expression, location information and 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 (3) 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.

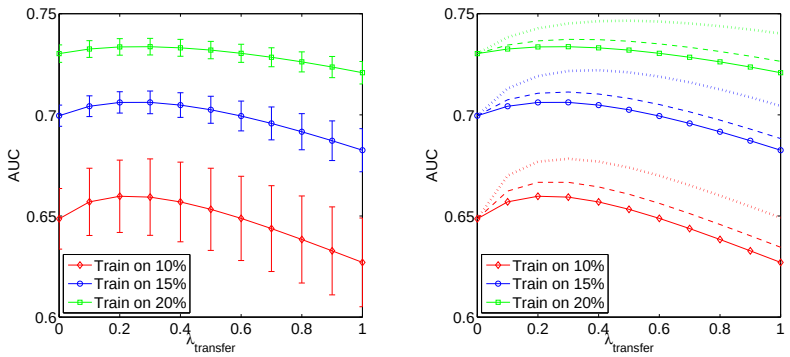

Fig. 1: Plots of the AUC values as a function of the parameter of the tranfer learning $\lambda_{transfer}$. Left: The three plots correspond to three sizes of the training data, 10%, 15% and 20%, the error bars give the standard deviation to the mean for the 10 runs. Right: The plots are an extension of the three plots from the left-hand side to multiple reference species: 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.

4 Discussions

We described a method for transfer learning which increases the training set of a target species using a converter from the output space of the reference species to

the output space of the target species. We conducted experiments using baker yeast as the target species. The experiments show that the transfer learning improves the performance, particularly with a smaller size of the training set. Furthermore, we see that considering multiple reference species increases the performance.

Transfer learning has been recently considered for predicting PPIs. The approach of [6] is directed to simulatanously learning PPI networks of multiple species in a setting different then ours, i.e., genomic data and PPIs are avaialble for all species, while we consider that genomic data is available only for the target species. [8] is another recent study where the some extra information is added in a semi-supervised multi-task learnings etting. The extra information is PPI that are so-called 'weakly labeled'

There are several directions for future research to consider. *i)* Methods which involve other cost terms for transferring the information from other species; *ii)* Other prediction function; for example, the prediction function used for string to string mapping in [3].

Acknowledgments We acknowledge support from the Netherlands Organization for Scientific Research.

References

1. B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.
2. C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 153–160, New York, NY, USA, 2005. ACM.
3. C. Cortes, M. Mohri, and J. Weston. A general regression framework for learning string-to-string mappings. In *Predicting Structured Data*. MIT Press, 2007.
4. T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.
5. P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc. Inferring biological networks with output kernel trees. *BMC Bioinformatics*, 2007.
6. 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.
7. R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the ICML*, 2002.
8. Y Qi, O Tastan, JG Carbonell, J Klein-Seetharaman, and J Weston. Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins. *Bioinformatics*, 26(18):i645–i652, 2010.
9. J.-P. Vert. Reconstruction of biological networks by supervised machine learning approaches. *arXiv:0806.0215v2*, 2008.
10. J. Weston, O. Chapelle, A. Elisseeff, B. Schlkopf, and V. Vapnik. Kernel dependency estimation. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing*, volume 15, pages 873–880, Cambridge, MA, USA, 2003. MIT Press.