On learning with kernels for unordered pairs

Martial Hue Martial.Hue@mines-paristech.fr
Jean-Philippe Vert Jean-Philippe.Vert@mines-paristech.fr
Mines ParisTech, Centre for Computational Biology, 35 rue Saint Honoré, F-77300 Fontainebleau, France
Institut Curie, F-75248, Paris, France,
INSERM, U900, Paris, F-75248, France

Abstract

We propose and analyze two strategies to learn over unordered pairs with kernels, and provide a common theoretical framework to compare them. The strategies are related to methods that were recently investigated to predict edges in biological networks. We show that both strategies differ in their loss function and in the kernels they use. We deduce in particular a smooth interpolation between the two approaches, as well as new ways to learn over unordered pairs. The different approaches are tested on the inference of missing edges in two biological networks.

1. Introduction

This work is motivated by a promising line of research that has attracted some interest recently, namely, using machine learning and in particular kernel methods to predict interactions in biological networks from genomic data (Bock & Gough, 2001; Yamanishi et al., 2004; Ben-Hur & Noble, 2005; Vert & Yamanishi, 2005; Kato et al., 2005; Martin et al., 2005; Bleakley et al., 2007). The problem is, given a list of genes or proteins known to interact or not, to predict whether new pairs interact or not. For that purpose, data about the individual proteins are available, such as their amino acid sequence or their expression levels across many experimental conditions. This problem has many applications in systems biology, since it could allow us to systematically increase our knowledge of complex biological processes from the wealth of data that can be produced by recent high-throughput technologies. Learning pairwise relationships has also many possible applications beyond biology, e.g., in sociology and marketing to predict social interactions between persons from data about individuals.

From a machine learning perspective, this problem can be formulated as a binary supervised classification problem over unordered pairs of genes (we only consider undirected interactions here). Let us denote by $\mathcal{A}$ the space to represent the individual proteins, typically a Hilbert space associated to a reproducing kernel. A difficulty that must be addressed, then, is that of representing an unordered pair of points $\{a, b\}$ from representations of individual points a and b in $\mathcal{A}$. In particular, while an ordered pair (a, b) is naturally an element of the product space $\mathcal{A} \times \mathcal{A}$,which inherits a Hilbert space structure from that of $\mathcal{A}$, the problem of *invariance* by permutation must be addressed when unordered pairs $\{a, b\}$ are considered. Some authors have proposed to use the natural representation or ordered pairs in the product space, either ignoring the invariance constraint (Bock & Gough, 2001) or enforcing it by symmetrizing the representation (Ben-Hur & Noble, 2005; Martin et al., 2005). Others have prefered to rephrase the problem as learning over ordered pairs to estimate a non-symmetric function, and enforce the invariance in the predictions by averaging *a posteriori* the predictions made on symmetric pairs (a, b) and (b, a) to make a prediction for the unordered pair $\{a, b\}$ (Bleakley et al., 2007).

Here we wish to clarify the theoretical relationships between these last two approaches, i.e., symmetrizing the representation of ordered pair before learning, or learning over ordered pairs and symmetrizing the prediction *a posteriori*. We show that, after a certain reformulation, they only differ in the loss function employed and in the representation for ordered pairs they use. This allows us to propose a whole family of new methods, and in particular to smoothly interpolate between both approaches.

Learning over unordered pairs from a description or ordered pairs is a particular instance of multi-instance

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

learning, where one learns over sets of points. Indeed, an unordered pair $\{a, b\}$ can be seen as a subset of two ordered pairs $\{(a, b), (b, a)\}$. Most results in this paper can be generalized to this more general settings of multiple instance (MI) learning (Dietterich et al., 1997), in particular one of the two strategies we investigate corresponds to the MI kernel proposed by Gärtner et al. (2002) in the context of MI learning. We therefore state below the formalization of the problem and the main results in the general context of MI learning, and deduce particular results when the equivalence classes are unordered pairs.

2. Setting and notations

Let $\mathcal{X}$ be a set endowed with a positive definite (p.d.) kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Much research in statistics and machine learning has been devoted to the inference of a function $f : \mathcal{X} \to \mathbb{R}$ from a set of n observations of input/output pairs $\mathcal{S} = (x_i, y_i)_{i=1,\ldots,n}$, where for each $i = 1, \ldots, n$, $x_i \in \mathcal{X}$ and $y_i \in \mathbb{R}$. In particular, a successful line of research has been to estimate a function f in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ associated with the kernel K as the solution of the following optimization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f(x_i)\right) + \lambda ||f||_{\mathcal{H}}^2 , \qquad (1)$$

where $\ell : \mathbb{R}^2 \to \mathbb{R}$ is a loss function convex in its second argument, $||f||_{\mathcal{H}}$ is the norm of f in the RKHS $\mathcal{H}$, and λ is a regularization parameter (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004).

In this paper we consider the situation where we infer a function not over $\mathcal{X}$, but over a set $\mathcal{P}$ of finite subsets of $\mathcal{X}$. This is the typical scenario of MI learning. Each element $p \in \mathcal{P}$ is a finite subset of $|p|$ points in $\mathcal{X}$, i.e., $p = \left\{x^1, \ldots, x^{|p|}\right\}$ with $x^i \in \mathcal{X}$ for $i = 1, \ldots, |p|$. We are given a series of n observations of input/output pairs $(p_i, y_i)_{i=1,\ldots,n}$, where for each $i = 1, \ldots, n$, $p_i \in \mathcal{P}$ and $y_i \in \mathbb{R}$. We note that, contrary to the classical situation presented in the previous paragraph, the kernel K is on $\mathcal{X}$ while the inference problem is on $\mathcal{P}$.

As explained in the introduction, this general MI learning formulation covers in particular the problem of learning over unordered pairs when a kernel over ordered pairs is available: $\mathcal{X}$ represents then the set of ordered pairs of the form $x = (a, b)$, i.e., $\mathcal{X} = \mathcal{A}^2$, where $\mathcal{A}$ is the set of individuals, while $\mathcal{P}$ is the set of pairs of ordered pairs obtained by permutation, i.e., $p = \{(a, b), (b, a)\}$. Elements of $\mathcal{P}$ can therefore be thought as unordered pairs. Figure 1 illustrates the representation.

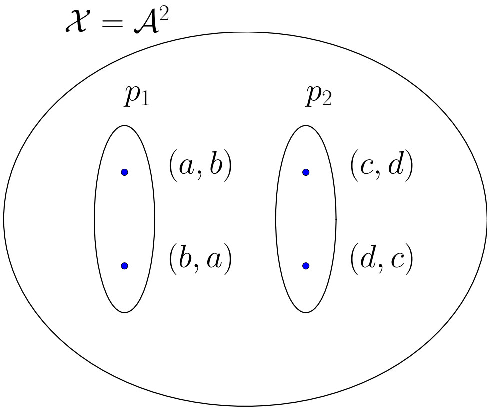

Figure 1. Illustration of our notations. $\mathcal{A}$ is the space of individual vertices (genes or proteins). $\mathcal{X} = \mathcal{A}^2$ is the set of ordered pairs of vertices. $\mathcal{P}$ is a partition of $\mathcal{X}$, where each element $p \in \mathcal{P}$ is a set of two symmetric ordered pairs, such as $p_1 = \{(a, b), (b, a)\}$ and $p_2 = \{(c, d), (d, c)\}$.

3. Two strategies to learn over $\mathcal{P}$

We investigate two strategies to learn over $\mathcal{P}$. The first approach is to define a p.d. kernel over $\mathcal{P}$ using the one existing over $\mathcal{X}$, and to solve the inference problem over $\mathcal{P}$ using a classical kernel method. The second is to transform the inference problem over $\mathcal{P}$ into an inference problem over $\mathcal{X}$, and to solve the latter using a classical kernel method with the p.d. kernel K. We now describe each approach in more detail, keeping the general formalism of MI learning.

Strategy 1 : Inference over $\mathcal{P}$ (pair kernel)

The first approach is to derive a kernel $K_{\mathcal{P}}$ over $\mathcal{P}$ from the kernel K over $\mathcal{X}$. In the case of pairs, this amounts to defining a kernel over unordered pairs from a kernel over ordered pairs, which we call a *symmetric pair kernel*. Among the kernels for sets of points that have been investigated in the past, we consider the popular average:

$$K_{\mathcal{P}}(p, p') = \frac{1}{|p| \cdot |p'|} \sum_{x \in p, x' \in p'} K(x, x') . \qquad (2)$$

$K_{\mathcal{P}}$ is a particular instance of a convolution kernel (Haussler, 1999) and is therefore p.d. over $\mathcal{P}$. It was proposed as a kernel for MI learning by Gärtner et al. (2002). We denote by $\mathcal{H}_{\mathcal{P}}$ its RKHS. We can then estimate a function $f_* : \mathcal{P} \mapsto \mathbb{R}$ from the examples $(p_i, y_i)_{i=1,\ldots,n}$ by solving the following optimization problem in $\mathcal{H}_\mathcal{P}$:

$$f_* = \arg\min_{f \in \mathcal{H}_\mathcal{P}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(p_i)) + \lambda \|f\|^2_{\mathcal{H}_\mathcal{P}} , \qquad (3)$$

where λ is a regularization parameter. This amounts to a classical machine learning method (1) with the MI kernel (2).

Strategy 2 : Inference over $\mathcal{X}$ (pair duplication)

The second strategy is to infer a function over $\mathcal{X}$, and transform it into a function that is invariant over each subset in $\mathcal{P}$ *a posteriori*. For that purpose, we first expand each training example (p, y) of $\mathcal{P} \times \mathcal{Y}$ into several pairs of $\mathcal{X} \times \mathcal{Y}$ by taking all points $x \in p$ and assigning them the label y. That is, if $p_i = \{x_i^1, \ldots, x_i^{n_i}\}$, then we expand the training set $\mathcal{S}$ over $\mathcal{P} \times \mathcal{Y}$ to make a new training set $\mathcal{S}'$ over $\mathcal{X} \times \mathcal{Y}$:

$$\mathcal{S}' = \{(x_i^j, y_i), i = 1, \ldots, n; j = 1, \ldots, n_i\} \subset \mathcal{X} \times \mathcal{Y} .$$

We note that when p represents an unordered pair $\{a, b\}$, the expansion amounts to *duplicate* it into two ordered pairs $x = (a, b)$ and $x' = (b, a)$, both labeled with the same label as p. We then infer a function over $\mathcal{X}$ from this training set by solving the problem

$$g_* = \arg\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L_2(g, p_i, y_i) + \lambda \|g\|^2_{\mathcal{H}} , \qquad (4)$$

where the loss function associated to a subset $p \in \mathcal{P}$ is expanded over its elements according to:

$$L_2(g, p, y) = \frac{1}{|p|} \sum_{x \in p} \ell(y, g(x)) . \qquad (5)$$

In practice, this means that we just estimate the function g_* using a classical kernel learning method (1) over $\mathcal{S}'$ (up to a weighting of the loss associated to each point inversely proportional to the size of the subset p it belongs to). Finally, we obtain a function over $\mathcal{P}$ by averaging the values g_* takes over the elements of each subset, i.e., the final predictor is $f_{g^*} : \mathcal{P} \to \mathbb{R}$ given by

$$\forall p \in \mathcal{P}, \quad f_{g_*}(p) = \frac{1}{|p|} \sum_{x \in p} g_*(x) . \qquad (6)$$

4. Relation to previous work

These two strategies cover several successful approaches that have been proposed to infer undirected interactions in biological networks, and which we now review. In this section we therefore consider the case where $\mathcal{X} = \mathcal{A}^2$ is the set of ordered pairs of elements of a set $\mathcal{A}$ (e.g., proteins), and $\mathcal{P}$ is the set of unordered pairs of the form $p = \{(a, b), (b, a)\}$ for $a, b \in \mathcal{A}$.

4.1. The tensor product pairwise kernel (TPPK) approach

Given a kernel $K_\mathcal{A}$ over the set of individuals $\mathcal{A}$, (Ben-Hur & Noble, 2005; Martin et al., 2005) define the following tensor product pairwise kernel between unordered pairs:

$$\begin{aligned} K_{TPPK}\left(\{a,b\},\{c,d\}\right) = & K_\mathcal{A}(a,c)K_\mathcal{A}(b,d) \\ & + K_\mathcal{A}(a,d)K_\mathcal{A}(b,c) . \end{aligned} \qquad (7)$$

Then they propose to infer a function over unordered pairs $\mathcal{P}$ by applying a kernel method over $\mathcal{P}$ with the TPPK kernel, i.e., by solving (3) with the kernel (7). We now state the formal equivalence of this approach with both strategies:

Proposition 1. *Let $\mathcal{X} = \mathcal{A}^2$ be endowed with the p.d. kernel:*

$$K\left((a,b),(c,d)\right) = 2K_\mathcal{A}(a,c)K_\mathcal{A}(b,d) . \qquad (8)$$

Then the TPPK approach is equivalent to both Strategy 1 *and* Strategy 2 *, in the sense that they all give the same solution.*

The equivalence between the TPPK approach and *Strategy 1* is a simple consequence of the following equality between the TPPK kernel and the kernel $K_\mathcal{P}$ defined over $\mathcal{P}$ in *Strategy 1* :

$$\begin{aligned} K_\mathcal{P}(\{a,b\},\{c,d\}) = & \frac{1}{4}[K((a,b),(c,d)) + K((a,b),(d,c)) \\ & + K((b,a),(c,d)) + K((b,a),(d,c))] \\ = & K_\mathcal{A}(a,c)K_\mathcal{A}(b,d) + K_\mathcal{A}(a,d)K_\mathcal{A}(b,c) \\ = & K_{TPPK}\left(\{a,b\},\{c,d\}\right) . \end{aligned}$$

The equivalence of the TPPK approach with *Strategy 2* is less obvious at first sight, and is proved in the Appendix. In practice, this equivalence between the TPPK approach and *Strategy 2* does not mean that we should solve the problem with *Strategy 2* , since the latter involves estimating the classifier from more training examples than the former. This equivalence is useful since it will allow us in section 6 to propose a smooth interpolation between the TPPK approach and another successful approach for inference over unordered pairs, the local model, reviewed below, by interpolating between kernels in *Strategy 2* .

4.2. The local model approach

To address the same problem of inferring edges in biological networks, (Bleakley et al., 2007) propose to learn independently local subnetworks and merge them for prediction. More precisely, for each individual $a \in \mathcal{A}$, its set of mates in the unordered pairs of the training set is collected. If a pair $\{a, b\}$ has the label y in the training set, then the mate b is given the label y. From the set of labeled mates, one estimates the function $g_a : \mathcal{A} \mapsto \mathbb{R}$ with a kernel method (using the kernel $K_{\mathcal{A}}$ on $\mathcal{A}$). g_a represents then the local subnetwork inferred near a, in the sense that $g_a(b) > 0$ means that b is predicted to interact with a. Finally, once the functions g_a are estimated for all $a \in \mathcal{A}$, we obtain a prediction $f_{local} : \mathcal{P} \mapsto \mathbb{R}$ for a new unordered pair $p = \{c, d\}$ by averaging the local models in c and d as follows:

$$f_{local}\left(\{c, d\}\right) = \frac{g_c(d) + g_d(c)}{2} .$$

Not surprisingly, this local approach is a particular case of *Strategy 2* with a particular kernel:

Proposition 2. *Let $\mathcal{X} = \mathcal{A}^2$ be endowed with the p.d. kernel:*

$$K\left((a, b), (c, d)\right) = \delta(a, c) K_{\mathcal{A}}(b, d) , \qquad (9)$$

where δ is the Kronecker kernel ($\delta(a, c) = 1$ if $a = c$, 0 otherwise). Then the local approach is equivalent to Strategy 2 *, in the sense that they give the same solution.*

The proof of this statement is postponed to the Appendix. It should be noted that, in contrast to the TPPK approach and the kernel (8), *Strategy 1* and *Strategy 2* are not equivalent with the kernel (9).

5. Comparison of *Strategy 1* and *Strategy 2*

In this Section we come back to the general MI learning setting of learning over a general set of subsets $\mathcal{P}$. While *Strategy 1* and *Strategy 2* are quite different in their presentation, the following results, whose proof is postponed to the Appendix, shows how to interpret them in a unified framework and highlight their similarity and difference.

Theorem 1. *The solution (3) of* Strategy 1 *is also the solution of* Strategy 2 *when the loss function (5) is replaced by:*

$$L_1(g, p, y) = \ell\left(y, \frac{1}{|p|}\sum_{x\in p} g(x)\right) . \qquad (10)$$

This results shows that the difference between *Strategy 1* and *Strategy 2* can be thought as a difference between the loss functions they use. More precisely, learning with a convolution kernel (2) over labeled sets in $\mathcal{P}$ is thus formally equivalent to expand the labeled sets into labeled points, learn over the labeled points with the loss function (10), and then make a prediction over a set as the average prediction over the points it contains. *Strategy 2* follows the same steps, but with a different loss function.

It is instructive to compare the losses L_1 and L_2 used by both strategies. Since the loss function $\ell(y, u)$ is convex in its second argument, we immediately get that:

$$L_1(g, p, y) \leq L_2(g, p, y) .$$

Hence L_2 is a surrogate of L_1, which implies that the solution found by *Strategy 2* , which has a small average L_2 loss, must also have a small average L_1 loss. In a sense, *Strategy 2* is more conservative than *Strategy 1* in the sense that it penalizes a bad prediction for *any* element of a set p, while *Strategy 1* only penalizes the average prediction over p. This suggests that *Strategy 2* may be better adapted to situations where one expects all individuals within a subset p to behave similarly, while *Strategy 1* may be better suited to problems where only a few elements of a subset are sufficient to characterize the class of the subset, e.g., in multi-instance learning (Gärtner et al., 2002).

Finally, Theorem 1 suggests that new algorithms may be investigated by modifying the loss function (5) employed in *Strategy 2* , besides L_1 and L_2. For example, in order to be even more conservative than L_2 and penalize the worse prediction within a subset instead, one could consider the following loss function:

$$L_\infty(g, p, y) = \max_{x\in p} \ell\left(y, g(x)\right)$$

We leave the investigation of this and other new formulations to future work.

6. Interpolating between the TPPK and local approaches

We have shown that the TPPK approach of (Ben-Hur & Noble, 2005; Martin et al., 2005) is equivalent to both *Strategy 1* and *Strategy 2* with the particular kernel (8) over ordered pairs, while the local approach of (Bleakley et al., 2007) is equivalent to *Strategy 2* with the kernel (9). While both approaches have been applied separately, this common framework provides opportunities to mix them by smoothly interpolating between them. More precisely, let us consider the following kernel on ordered pairs, for $\lambda \in [0, 1]$:

$$K^\lambda\left((a, b), (c, d)\right) = \left[\lambda\delta(a, c) + (1-\lambda)K_{\mathcal{A}}(a, c)\right] \cdot K_{\mathcal{A}}(b, d) . \qquad (11)$$

Using K^0 with *Strategy 1* and *Strategy 2* gives rise to the TPPK approach. Using K^1 with *Strategy 2* is the local approach. Varying λ between 0 and 1, and using K^{λ} with *Strategy 2* therefore smoothly interpolates between the TPPK and local approaches. Using K^{λ} with *Strategy 1* for $0 \leq \lambda \leq 1$ provides a new set of solutions, starting from the TPPK solution for $\lambda = 0$, which combine the less conservative loss function L_1 with a kernel less symmetric than TPPK. In terms of computation requirements, *Strategy 1* requires solving a pattern recognition problem with n training examples, while *Strategy 2* solves a similar problem with $2n$ examples. *Strategy 1* with K^1 (the local approach) benefits from an important computational advantage, since the learning problem decomposes as local and uncoupled problems which can be solved separately. If the n training pairs involve m proteins, each present in n/m pairs, then the local model must solve m problems with n/m points. Since the complexity of learning is typically at least quadratic in the number of points, we typically gain a factor of m over *Strategy 1* with other kernels.

7. Experiments

To evaluate the performance of the different formulations we test them on two benchmark biological networks used in previous studies (Yamanishi et al., 2004; Vert & Yamanishi, 2005; Kato et al., 2005; Bleakley et al., 2007). Both networks represent graphs of proteins of the budding yeast *S. cerevisiae*. The first network is the *metabolic network*, which connects pairs of enzymes which catalyse successive reactions in the known biochemical networks. This graph contains 668 vertices and 2782 undirected edges. The second network is the protein-protein interaction (PPI) network of (von Mering et al., 2002) where only high-confidence interactions are kept. This graph contains 984 vertices and 2438 edges. To predict the edges in both network, we use several kernels between proteins (vertices), which encode various biological information. For the metabolic network, we use three basic kernels based on expression data, cellular localization, and phylogenetic profiles, as well as a kernel that combines them as a sum. For the PPI network, we used four kernel based on expression data, localization, phylogenetic profiles, and yeast two-hybrid experiments, as well as their combination with a sum. All data are available from the supplementary information of (Yamanishi et al., 2004) [1].

For each network, we randomly generated nine times as many negative pairs (decoys) as known edges. We then performed a 5-fold cross-validation classification experiment to discriminate true edges from decoys with a SVM, using the *libsvm* implementation with a custom kernel. For each split, the regularization parameter of the SVM (C) was chosen on the grid $\{1, 2, 4, 8, 16, 32\}$ by maximizing the mean area under the ROC curve (AUC) on an internal 5-fold cross-validation on the training set. Using this common procedure, we compared the various kernels K^{λ} in (11) for 11 values of λ uniformly spaced in the interval $[0, 1]$, and the two approaches *Strategy 1* and *Strategy 2* .

Table 1 shows, for each of the 9 experiments, which configuration reaches the best performance. The main message is that there is no clear winner, and that it may therefore be useful to consider various strategies for a given problem. More precisely, for six datasets, *Strategy 2* performs equally as or better than *Strategy 1* uniformly on $\lambda \in [0, 1]$, while *Strategy 1* performs equally as or better than *Strategy 2* in the remaining three datasets. In five cases, the maximal AUC is attained at λ strictly between 0 and 1. In these cases, the improvement over $TPPK$ and the local model is significant for the experiments "interaction, expression" and "metabolic, expression", corresponding to cases where the newly proposed interpolation significantly outperforms both existing methods, with a p-value $< 10^{-3}$.

Figures 2 shows three cases of the various patterns we can observe when we investigate the performance of both strategies as a function of λ. As expected, the performance of *Strategy 1* and *Strategy 2* coincide at $\lambda = 0$, which corresponds to the TPPK approach. The relative performance of both strategies, and their performance as a function of λ greatly depends on the dataset.

Table 1. Strategy and kernel realizing the maximum mean AUC for nine metabolic and protein-protein interaction networks experiments, with the kernel K^{λ} for $\lambda \in [0, 1]$.

benchmark	best kernel
interaction, exp	Duplicate, $\lambda = 0.7$
interaction, loc	Pair kernel, $\lambda = 0.6$
interaction, phy	Duplicate, $\lambda = 0.8$
interaction, y2h	Duplicate / Pair kernel, $\lambda = 0$
interaction, integrated	Duplicate / Pair kernel, $\lambda = 0$
metabolic, exp	Pair kernel, $\lambda = 0.6$
metabolic, loc	Pair kernel, $\lambda = 1$
metabolic, phy	Pair kernel, $\lambda = 0.6$
metabolic, integrated	Duplicate / Pair kernel, $\lambda = 0$

8. Conclusion

We proposed a theoretical analysis of two strategies that were proposed recently to learn over pairs. We

[1] Available at http://web.kuicr.kyoto-u.ac.jp/~yoshi/ismb04

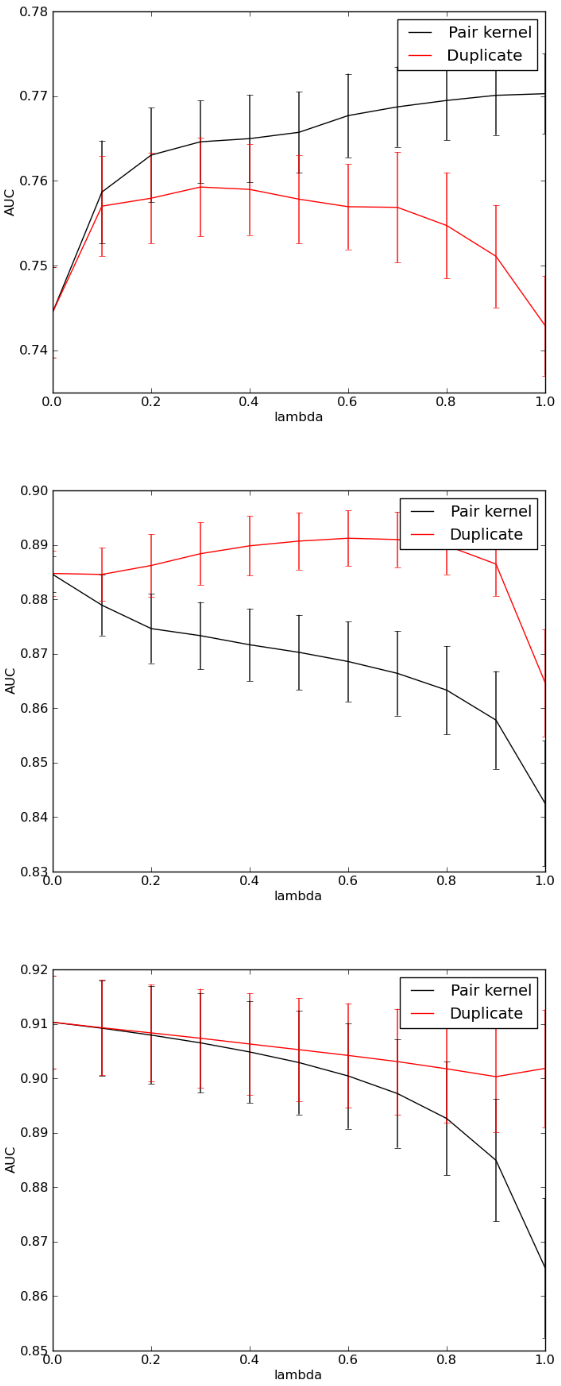

Figure 2. Mean AUC for the metabolic, loc (up), protein-protein interaction, exp (middle) and interaction, integrated (down) networks experiment, with the kernel K^{λ} for $\lambda \in [0, 1]$. The red curves (Duplication) correspond to *Strategy 2* , while the black curves (Pair kernel) correspond to *Strategy 1* .

have shown that they can be compared in terms of loss function, and in terms of kernel used to represent a directed pair. We have derived from this analysis a family of new methods which interpolate between both methods, and have shown on real data that the different strategies can be relevant on different datasets. The problem of automatically finding the good strategy and the good parameters for a given dataset remains to be investigated, as well as the relevance of the methods to more general MI learning problems.

Appendix

Proof of Proposition 1

To show the equivalence between the TPPK approach and *Strategy 2* with the product kernel (8), let us first rewrite the TPPK approach as an optimization problem. Let $\mathcal{H}_{\mathcal{A}}$ be a Hilbert space and $\Phi_{\mathcal{A}} : \mathcal{A} \to \mathcal{H}_{\mathcal{A}}$ be such that, for all $a, b \in \mathcal{A}$, $K_{\mathcal{A}}(a, b) = \langle \Phi_{\mathcal{A}}(a), \Phi_{\mathcal{A}}(b) \rangle_{\mathcal{H}_{\mathcal{A}}}$. Then we can express a product kernel as an inner product in $\mathcal{H}_2 = \mathcal{H}_{\mathcal{A}} \otimes \mathcal{H}_{\mathcal{A}}$ as follows:

$$K_{\mathcal{A}}(a, c) K_{\mathcal{A}}(b, d) = \langle \Psi_1(a, b), \Psi_1(c, d) \rangle_{\mathcal{H}_2} \, , \qquad (12)$$

where $\Psi_1 : \mathcal{X} \to \mathcal{H}_2$ is defined by $\Psi_1(a, b) = \Phi_{\mathcal{A}}(a) \otimes \Phi_{\mathcal{A}}(b)$. For any $\{a, b\} \in \mathcal{P}$, let now:

$$\Psi_2\left(\{a, b\}\right) = \frac{\Psi_1(a, b) + \Psi_1(b, a)}{\sqrt{2}} \, . \qquad (13)$$

By definition of the TPPK kernel (7) we easily see that it can be rewritten as an inner product in $\mathcal{H}_2$:

$$K_{TPPK}\left(\{a, b\}, \{c, d\}\right) = \langle \Psi_2(\{a, b\}), \Psi_2(\{c, d\}) \rangle_{\mathcal{H}_2} \, .$$

We therefore deduce that the TPPK method can be expressed as estimating the function $p \in \mathcal{P} \mapsto f_*(p) = \langle W_*, \Psi_2(p) \rangle_{\mathcal{H}_2}$, where W_* solves the following optimization problem:

$$\min_{W \in \mathcal{H}_2} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, \langle W, \Psi_2(p_i) \rangle_{\mathcal{H}_2}\right) + \lambda ||W||^2_{\mathcal{H}_2} \, . \qquad (14)$$

Let us now show that *Strategy 2* with the product kernel (8) estimates the same function. Let $\Psi_0 = \sqrt{2}\Psi_1$. Then, by (12), we can express the kernel (8) as:

$$2K_{\mathcal{A}}(a, c) K_{\mathcal{A}}(b, d) = \langle \Psi_0(a, b), \Psi_0(c, d) \rangle_{\mathcal{H}_2} \, .$$

Therefore we can express the inference in *Strategy 2* as an optimization problem in $\mathcal{H}_2$:

$$\begin{aligned} \min_{W \in \mathcal{H}_2} \frac{1}{n} \sum_{i=1}^{n} [& \frac{\ell\left(y_i, \langle W, \Psi_0(a_i, b_i) \rangle_{\mathcal{H}_2}\right)}{2} \\ & + \frac{\ell\left(y_i, \langle W, \Psi_0(b_i, a_i) \rangle_{\mathcal{H}_2}\right)}{2}] \\ & + \lambda ||W||^2_{\mathcal{H}_2} \, . \end{aligned} \qquad (15)$$

For any $W \in \mathcal{H}_2$, seen as a Hilbert-Schmidt operator by isometry, let $W' \in \mathcal{H}_2$ be its adjoint, characterized by $\langle W, u \otimes v \rangle_{\mathcal{H}_2} = \langle W', v \otimes u \rangle_{\mathcal{H}_2}$. This implies, for any $a, b \in \mathcal{A}$,

$$\langle W, \Psi_0(a,b) \rangle_{\mathcal{H}_2} = \langle W', \Psi_0(b,a) \rangle_{\mathcal{H}_2} \,. \qquad (16)$$

We now show that the solution of (15) is self-adjoint, i.e., satisfies $W = W'$. For this we show that for any $W \in \mathcal{H}_2$, $(W + W')/2$ reaches a value strictly smaller than W in the objective function of (15) if $W \neq W'$. Indeed, by convexity of ℓ and (16) we first get, for any $a, b \in \mathcal{A}$:

$$\ell\left(y_i, \left\langle \frac{W+W'}{2}, \Psi_0(a,b) \right\rangle_{\mathcal{H}_2}\right) \leq \frac{\ell\left(y_i, \langle W, \Psi_0(a,b)\rangle_{\mathcal{H}_2}\right) + \ell\left(y_i, \langle W, \Psi_0(b,a)\rangle_{\mathcal{H}_2}\right)}{2} \,.$$

Plugging this into (15) we see that for any $W \in \mathcal{H}_2$, the first term of the objective function is at least as small for $(W + W')/2$ as for W. Now, since $||W||_{\mathcal{H}_2} = ||W'||_{\mathcal{H}_2}$, and by strict convexity of the Hilbert norm, we see that the second term of the objective function is strictly smaller for $(W + W')/2$ than for W except if $W = W'$. This proves that the solution of (15) is self-adjoint.

Since the W solution of (15) is self-adjoint, it satisfies in particular, for any $a, b \in \mathcal{A}$:

$$\begin{aligned} \langle W, \Psi_0(a,b) \rangle_{\mathcal{H}_2} &= \langle W, \Psi_0(b,a) \rangle_{\mathcal{H}_2} \\ &= \left\langle W, \frac{\Psi_0(b,a) + \Psi_0(a,b)}{2} \right\rangle_{\mathcal{H}_2} \\ &= \langle W, \Psi_2(\{a,b\}) \rangle_{\mathcal{H}_2} \,, \end{aligned} \qquad (17)$$

where the last equality is obtained from the definitions $\Psi_0 = \sqrt{2}\Psi_1$ and (13). Plugging this equality into (15) and rearranging the terms, we see that we recover exactly (14). Therefore the solution $W \in \mathcal{H}_2$ found by the TPPK approach and by *Strategy 2* with the kernel (8) are the same. Moreover, (17) shows that the predictions made by the two approaches on any pair are also identical. □

Proof of Proposition 2

In *Strategy 2* we expand each training undirected pair $p = \{a, b\}$ into two directed pairs (a, b) and (b, a), each labeled with the label of the original directed pair. Let us rename the elements of the resulting training set as $((u_i, v_i), y_i), i = 1, \ldots, 2n$. *Strategy 2* then solves the problem:

$$g_* = \arg\min_{g \in \mathcal{H}} \frac{1}{2n} \sum_{i=1}^{2n} \ell(g(u_i, v_i), y_i) + \lambda \|g\|_{\mathcal{H}}^2 \,. \qquad (18)$$

By the representer theorem, the solution g_* belongs to the following linear subspace of $\mathcal{H}$:

$$F = \operatorname{span}(\{K((u_i, v_i), \cdot), i = 1, \ldots, 2n\}) \,.$$

Let now $\mathcal{U}$ be the set of values taken by the first members $u_i, i = 1, \ldots, 2n$, and for $u \in \mathcal{U}$,

$$F_u = \operatorname{span}(\{K((u_i, v_i), \cdot) | u_i = u, i = 1, \ldots, 2n\}) \,.$$

Since $K((u, v), (u', v')) = 0$ for $u \neq u'$, we see that F_u is orthogonal to $F_{u'}$ for $u \neq u'$ and therefore $F = \bigoplus_{u \in \mathcal{U}} F_u$. For any $g \in F$, if we denote by g_u its projection onto F_u for $u \in \mathcal{U}$, this implies

$$\|g\|_{\mathcal{H}}^2 = \sum_{u \in \mathcal{U}} \|g_u\|_{\mathcal{H}}^2 \,.$$

Moreover, since $g_u(u', v') = 0$ for $u \neq u'$, we also have $g(u, v) = g_u(u, v)$ for any $u, v \in \mathcal{U}$. This implies that the optimization problem (18) decouples as follows:

$$g_* = \arg\min_{g \in F} \sum_{u \in \mathcal{U}} \left[\frac{1}{2n} \sum_{\substack{1 \leqslant i \leqslant 2n \\ u_i = u}} \ell(g_u(u, v_i), y_i) + \lambda \|g_u\|_{\mathcal{H}}^2 \right] . \qquad (19)$$

Each g_u can be found, independently from the others, by solving the subproblem in brackets using only the training pairs that start with u. Since F_u is isometric to the RKHS $\mathcal{H}_a$ of $K_{\mathcal{A}}$ by the mapping $g \in F_u \mapsto h \in \mathcal{H}_a$ with $h(v) = g_u(u, v)$, we see that g_{*u} is exactly the function found by the local model approach to infer the local subnetwork around u. Finally, we note that the prediction of *Strategy 2* for a new pair is given by:

$$\frac{g_*(u, v) + g_*(v, u)}{2} = \frac{g_{*u}(u, v) + g_{*v}(v, u)}{2} \,,$$

that is, exactly the same as the prediction of the local model. □

Proof of Theorem 1

For any function $g : \mathcal{X} \to \mathbb{R}$, remember that we denote by f_g the function $\mathcal{P} \to \mathbb{R}$ defined by

$$f_g(p) = \frac{1}{|p|} \sum_{x \in p} g(x) \,.$$

We first need the following results, which relates the norm in the RKHS $\mathcal{H}_{\mathcal{P}}$ of the convolution kernel $K_{\mathcal{P}}$ over $\mathcal{P}$ to the norm in the RKHS $\mathcal{H}$ of the original kernel K over $\mathcal{X}$.

Lemma 1. *For any function $f \in \mathcal{H}_{\mathcal{P}}$, it holds that*

$$\|f\|_{\mathcal{H}_{\mathcal{P}}} = \min\{\|g\|_{\mathcal{H}} \,:\, g \in \mathcal{H} \text{ and } f_g = f\} \,. \qquad (20)$$

We prove Lemma 1 in the next subsection. Plugging (20) into (3) we observe that the function $f_* \in \mathcal{H}_\mathcal{P}$ which minimizes the criterion (3) is equal to f_{g_*}, where $g_* \in \mathcal{H}$ solves the problem:

$$g_* = \arg\min_{g\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_g(p_i)) + \lambda \|g\|^2_\mathcal{H} \,. \tag{21}$$

Since $\ell(y_i, f_g(p_i))$ is exactly the loss $L_1(g, p_i, y_i)$ defined in (10), this concludes the proof of Theorem 1. □

Proof of Lemma 1

Let $\Phi : \mathcal{P} \to \mathcal{H}$ be defined for any $p \in \mathcal{P}$ by:

$$\Phi(p) = \frac{1}{|p|}\sum_{x\in p} K(x,\cdot)\,,$$

where $K(x,\cdot)$ denotes the function $t \mapsto K(x,t)$ in $\mathcal{H}$. For any $p, p' \in \mathcal{P}$ we have:

$$\begin{aligned}\langle\Phi(p),\Phi(p')\rangle_\mathcal{H} &= \frac{1}{|p|\cdot|p'|}\sum_{x\in p, x'\in p'} \langle K(x,\cdot)\,,\,K(x',\cdot)\rangle_\mathcal{H}\\ &= \frac{1}{|p|\cdot|p'|}\sum_{x\in p, x'\in p'} K(x,x')\\ &= K_\mathcal{P}(p,p')\,.\end{aligned}$$

If we denote by $E \subset \mathcal{H}$ the closure of the linear subspace of $\mathcal{H}$ spanned by $\{\Phi(p)\,,\, p \in \mathcal{P}\}$, this shows that E (endowed with the inner product inherited from $\mathcal{H}$) is isomorphic to $\mathcal{H}_\mathcal{P}$, since these are two Hilbert spaces spanned by $\mathcal{P}$ (respectively through $\Phi(p)$ and $K_\mathcal{P}(p,.)$) whose inner products coincide. Let now $f = \sum_i \alpha_i K_\mathcal{P}(p_i,.)$ be an element of $\mathcal{H}_\mathcal{P}$ (here the sum may be a convergent series), and let $h = \sum_i \alpha_i \Phi(p_i)$ be its image in E by the isometry between $\mathcal{H}_\mathcal{P}$ and E. Denoting by $\Pi_E : \mathcal{H} \to E$ the orthogonal projection onto E, we obtain by isometry:

$$\|f\|_{\mathcal{H}_\mathcal{P}} = \|h\|_\mathcal{H} = \min\left\{\|g\|_\mathcal{H} \,:\, g\in\mathcal{H},\, \Pi_E(g) = h\right\}\,. \tag{22}$$

The functions $g \in \mathcal{H}$ such that $\Pi_E(g) = h$ are characterized by the equalities $\langle g - h, \Phi(p)\rangle = 0$ for all $p \in \mathcal{P}$ or, equivalently:

$$\begin{aligned}\langle g,\Phi(p)\rangle &= \langle h,\Phi(p)\rangle\\ &= \sum_i \alpha_i \,\langle\Phi(p_i),\Phi(p)\rangle\\ &= \sum_i \alpha_i K_\mathcal{P}(p_i,p) = f(p)\,.\end{aligned} \tag{23}$$

On the other hand, by definition of $\Phi(p)$, we also have for all $p \in \mathcal{P}$:

$$\langle g,\Phi(p)\rangle = \frac{1}{|p|}\sum_{x\in\mathcal{P}}\langle g, K(x,.)\rangle = \frac{1}{|p|}\sum_{x\in\mathcal{P}} g(x) = f_g(p)\,. \tag{24}$$

Combining (23) and (24), we deduce that $g \in \mathcal{H}$ satisfies $\Pi_E(g) = h$ if and only if, for all $p \in \mathcal{P}$, $f_g(p) = f(p)$, that is, if and only if $f_g = f$. Combining this with (22) finishes the proof of Lemma 1. □

References

Ben-Hur, A. and Noble, W. S. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21 (Suppl. 1):i38–i46, Jun 2005.

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

Bock, J. R. and Gough, D. A. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17 (5):455–460, 2001.

Dietterich, T.G., Lathrop, R.H., and Lozano-Perez, T. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

Gärtner, T., Flach, P.A., Kowalczyk, A., and Smola, A.J. Multi-Instance Kernels. In Sammut, C. and Hoffmann, A. (eds.), *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 179–186. Morgan Kaufmann, 2002.

Haussler, D. Convolution Kernels on Discrete Structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz, 1999.

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

Martin, S., Roe, D., and Faulon, J.-L. Predicting protein-protein interactions using signature products. *Bioinformatics*, 21(2):218–226, Jan 2005.

Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, 2002.

Shawe-Taylor, J. and Cristianini, N. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA, 2004.

Vert, J.-P. and Yamanishi, Y. Supervised graph inference. In Saul, L. K., Weiss, Y., and Bottou, L. (eds.), *Adv. Neural Inform. Process. Syst.*, volume 17, pp. 1433–1440. MIT Press, Cambridge, MA, 2005.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, May 2002.

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