Protein-protein interaction Network Inference with Regularized Input and Output Kernel Methods

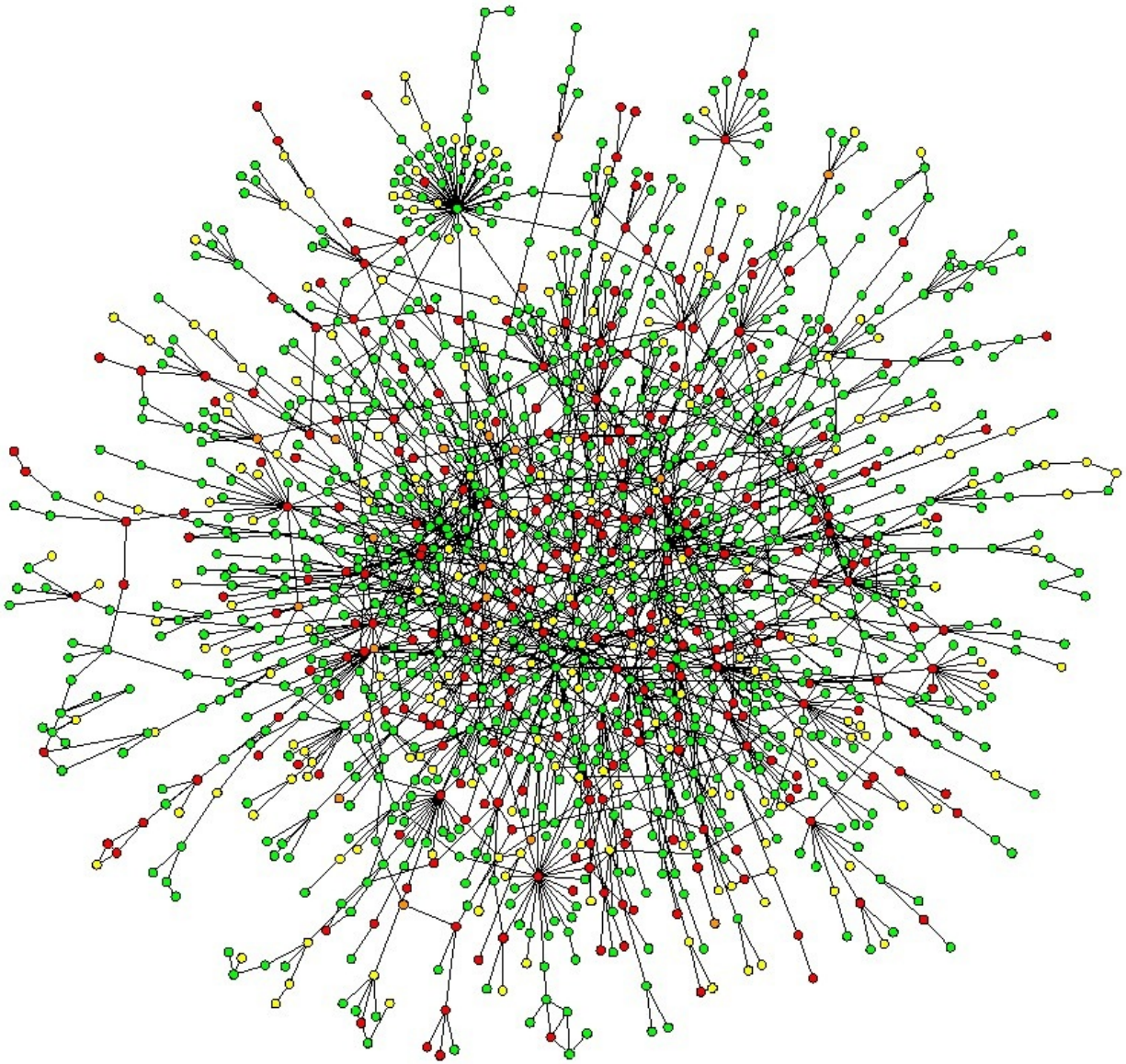
Florence d'Alché–Buc

IBISC CNRS 31 90, Université d'Evry, Genopole, Universud,France
Email: florence.dalche@ibisc.fr
MLSB 2010, Oct 16, Edinburgh

A few words about us

- **Evry**: 30km from Paris, city of genes, place for Genopole
- **University of Evry** belongs to UniverSud (universities of the south of Paris)
- **Our lab IBISC**: Computer Science and Complex Systems = 110 persons
- **Our group**: Machine learning with applications to Systems Biology (amisbio.ibisc.fr)
  - Modeling and learning dynamical biological systems
  - Mining structured data

Protein-protein interaction network in yeast



In silico ppi prediction methods

Motivation

- small scale experimental methods (protein-arrays, co-immunoprecipitations, FRET, NMR) are costly
- large scale systems like *Y2H* are known to produce high false positive rate

Goal

- Suggest interaction to be validated by low scale methods
- Prediction of protein-protein interactions (output ?) from known properties of proteins (input ?)

Data

Known features of proteins (Qi 2008)

- over-represented domains or motifs pairs
- phylogenetic signatures
- structural information, sequences
- co-expression of genes
- conservation of pairs of sequences

Protein-protein interactions

- Positive examples mainly
- A very few examples may be considered as negative

Existing approaches

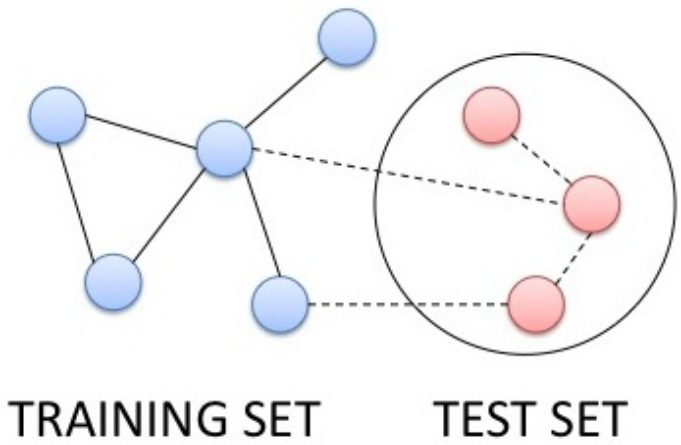
- Supervised edge inference or link prediction
- Semi-supervised and transductive learning

Supervised edge inference or link prediction

- o, o': two proteins
- $x(o)$ and $x(o')$: input feature vectors encoding some properties of o and o'

Decision function

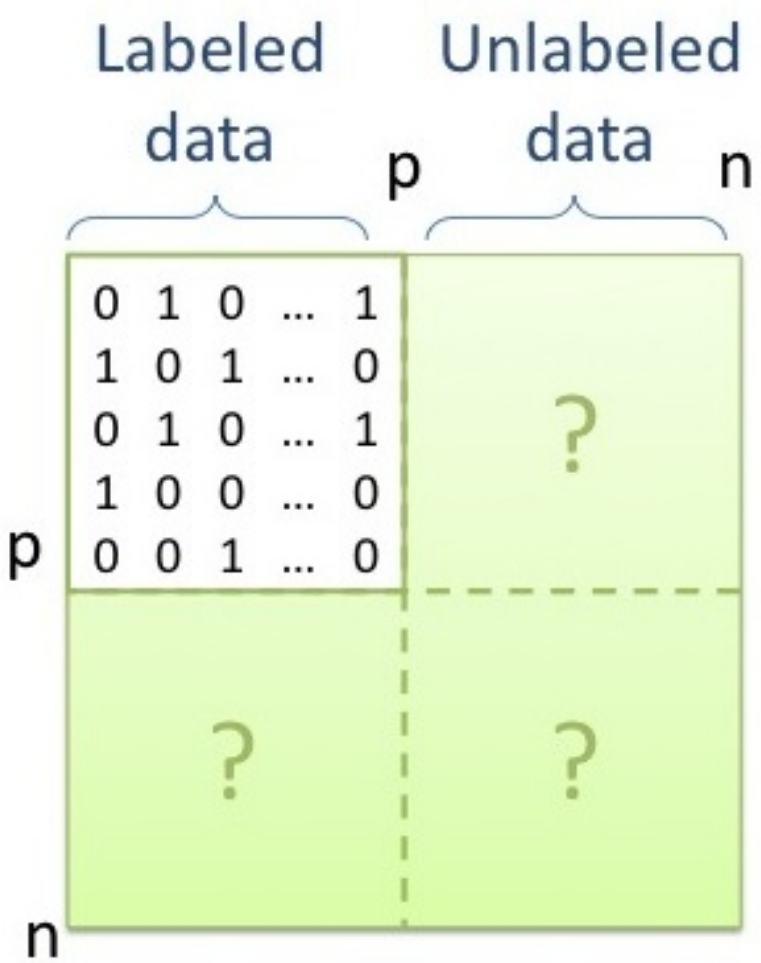
Learn a function $f : (x(o), x(o')) \rightarrow \{0, 1\}$ from
$\mathcal{S} = \{x(o_i), i = 1...p; W_{ij}, i, j = 1...p\}$



Supervised approaches

- Pairwise SVM [Ben-Hur and Noble 2005]
- Random forest, mixture of feature experts (Qi 2008)
- Supervised Learning of a kernel or a similarity
  - With KCCA [Yamanishi et al. 2004], with metric learning [Yamanishi and Vert 2005]
  - With output kernel regression tree [Geurts et al. 2006,07], with output kernel gradient boosting [Geurts et al. 2007]
- Supervised classification linked to a node
  - Another kind of supervised inference : local classifiers [Bleakley et al. 2007]

Supervised task or Matrix completion ?



Semi-supervised and transductive learning

- Kernel Matrix completion
  - Using EM [Tsuda et al. 2003] and [Kato et al. 2005]
  - Kernel Matrix Regression [Yamanishi and Vert 2007]
- Transductive or semi-supervised learning
  - Link Propagation [Kashima et al. 2009]
  - Mixture of Wishart Matrices [Dit-Yeung 2009]
  - Training set expansion [Yip and Gerstein 2009]

Aims

Build a general framework for link prediction that:

- Avoids working on pairs of objects (because computational complexity and i.i.d. assumption not met in that case)
- Is appropriate for a set of realistic variants of link prediction tasks
- Allows data integration and structured features encoding

Outline

Building a classifier f by learning a similarity k_y

- o: protein
- $x(o)$: input feature vector encoding some properties of o
- k_y: similarity between two proteins as nodes in the known graph

Similarity-based model

$$f_\theta(x(o), x(o')) = sgn(k_y(o, o') - \theta) \quad (1)$$

- Learning a proxy of k_y and choosing θ = learning the classifier f_θ

Building the classifier f by learning a kernel k_y

Output Kernel Regression: learn the feature map

Additional assumption: let k_y be a positive definite kernel and $y : \mathcal{O} \to \mathcal{Y}$ the feature map associated, $\mathcal{Y}$ the Hilbert space endowed with k_y as dot product:

$$k_y(o, o') = y(o)^T y(o') \tag{2}$$

Then instead of learning a kernel function k_y defined on pairs of objects, we can learn to predict y with a function h defined on $\mathcal{X} \to \mathcal{Y}$.

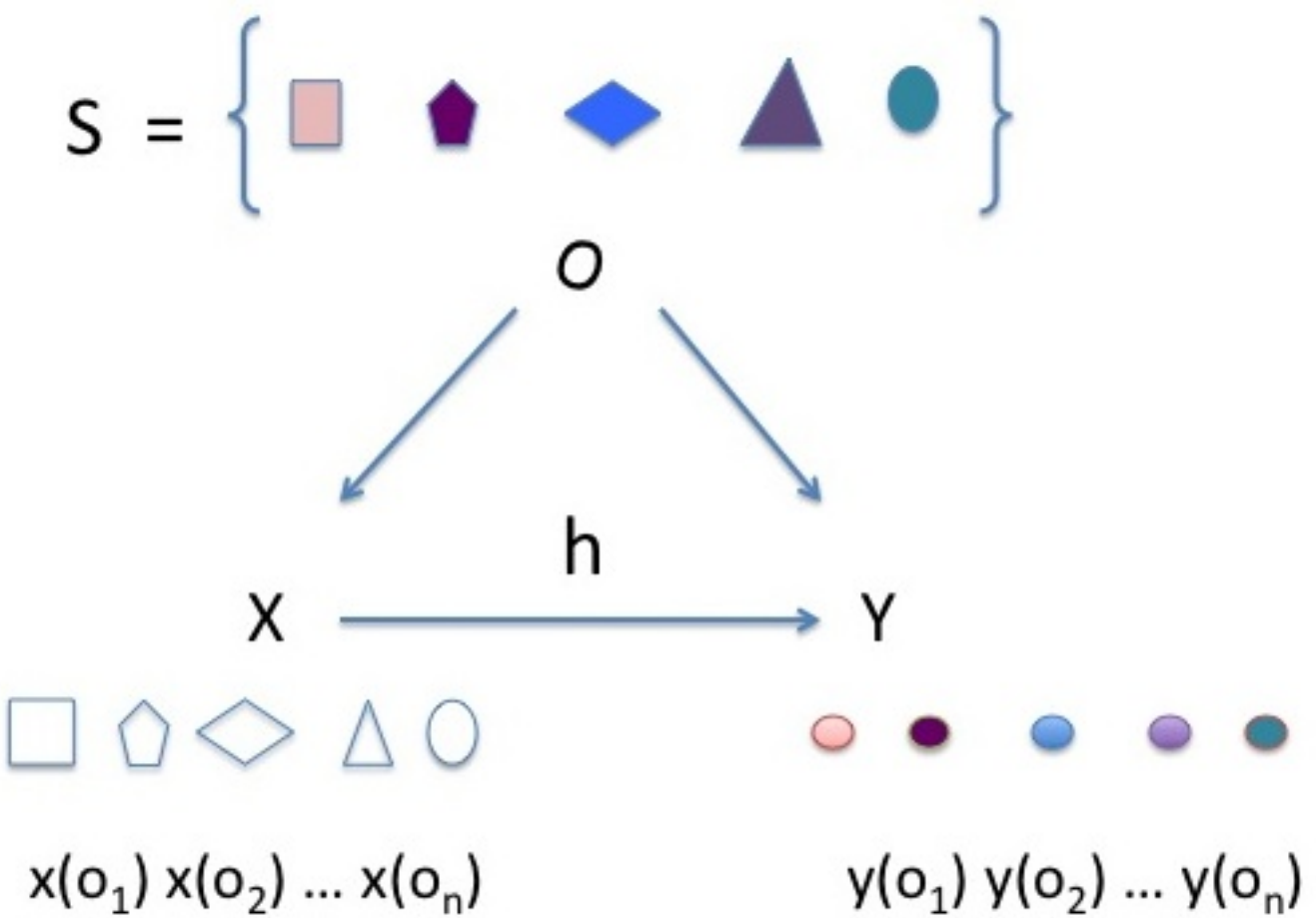
$$\hat{y}(o) = h(x(o)) \tag{3}$$

A new learning task

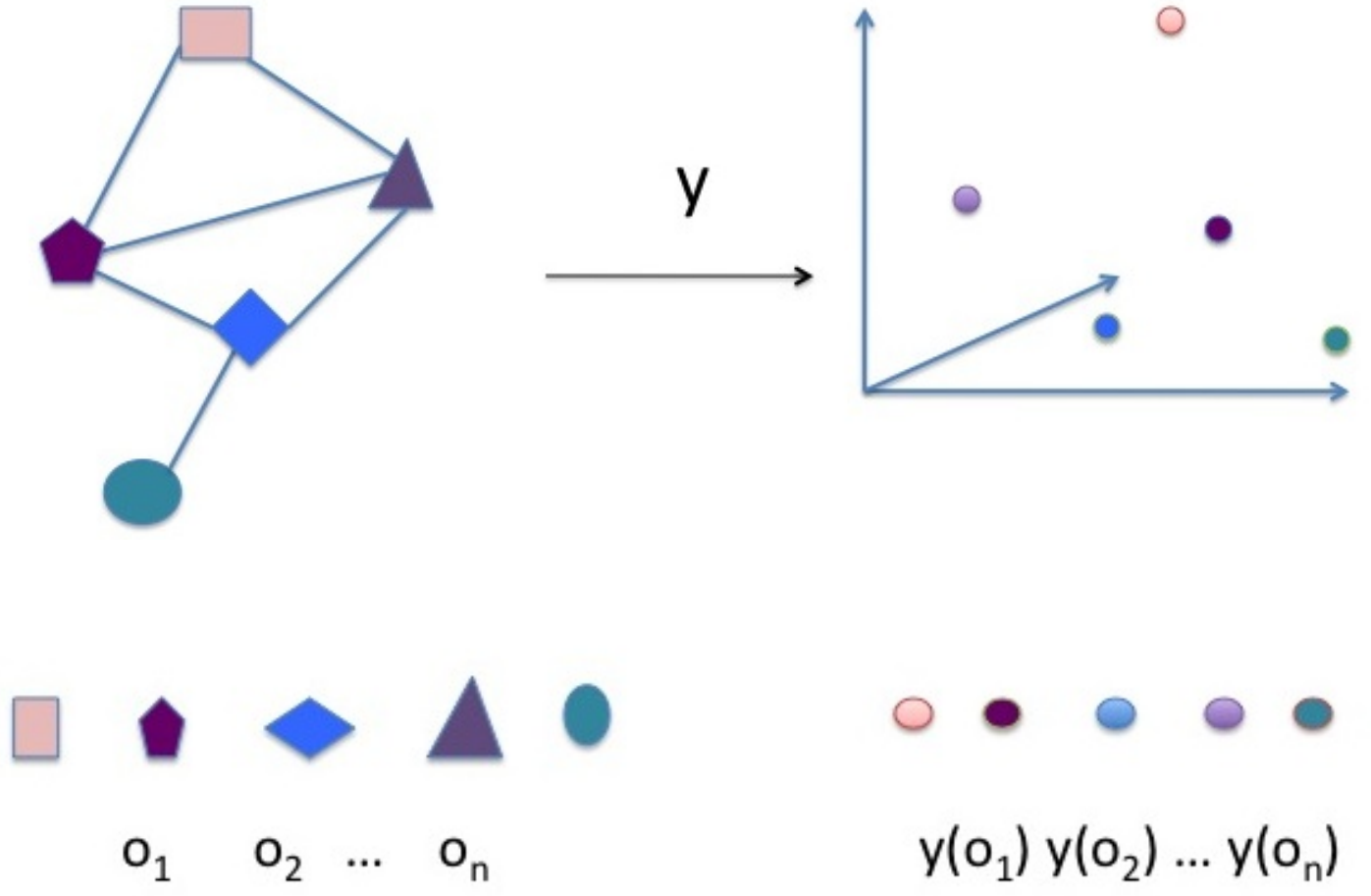
Output Kernel Regression

- if $\hat{y}(o) \in \mathcal{Y}$ and $\hat{y}(o') \in \mathcal{Y}$, then $\hat{k_y}(o, o') = \hat{y}(o)^T \hat{y}(o')$ is by construction a positive definite kernel
- if $\hat{y}(\cdot)$ is a good approximation of $y(\cdot)$ then $\hat{k_y}(o, o')$ is a good approximation of $k_y(o, o')$.
- (Geurts, Wehenkel and d'Alche–Buc 2006 and 2007)

Output Kernel Regression



Using the kernel trick for the outputs



Which k_y for our problem? a kernel on graph nodes

- Diffusion kernel (Kondor and Lafferty, 2002):
  The Gram matrix K with $K_{i,j} = k(o_i, o_j)$ is given by:

$$K = \exp(-\beta L)$$

  where the graph Laplacian L is defined by:

$$L = D - W$$

  if W is the adjacency matrix and D is the diagonal matrix of vertices degrees

$$L_{i,j} = \begin{cases} d_i & \text{the degree of node } o_i \text{ if } i = j; \\ -1 & \text{if } o_i \text{ and } o_j \text{ are connected;} \\ 0 & \text{otherwise.} \end{cases}$$

Supervised Output Kernel Regression

- For a given set $\{o_i, i = 1, \ldots, p\}$ from $\mathcal{O}$, given
  - an input representation: $x_i = x(o_i) \in \mathcal{X}$, the euclidean space or any kernel induced Hilbert space
  - an output representation: $K_y(i,j) = y(o_i)^T y(o_j)$ where $y \in \mathcal{Y}$, an Hilbert space endowed with a kernel k_y whose values are only known on dataset $\mathcal{D}$
  - a given class of functions H

Find a function $h : \mathcal{X} \to \mathcal{Y} \in H$ that minimizes the expectation of some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to I\!R$ over the joint distribution of input/output pairs:

$$E_{x,y}\{\ell(h(x), y)\}$$

What loss function ℓ to be chosen ? Which model ?

- The square loss $\ell(h(x), y) = \| h(x) - y \|_y^2$ only requires computation of dot products
- We only know K_y, thus search for h with outputs $\in$ span$(y_1, ..., y_i)$

Which family of models ?

- H_{OK3} : Output Kernel Regression Trees$*$
- Extension to bagging, boosting and random forests$*$
- H_{K2R2} : Input Output Kernel Regularized Regression $**$
- ....

$*$: joint work with Pierre Geurts and Louis Wehenkel
$**$: joint work with Céline Brouard (PhD student) and Marie Szafranski, many discussions with Pierre Geurts

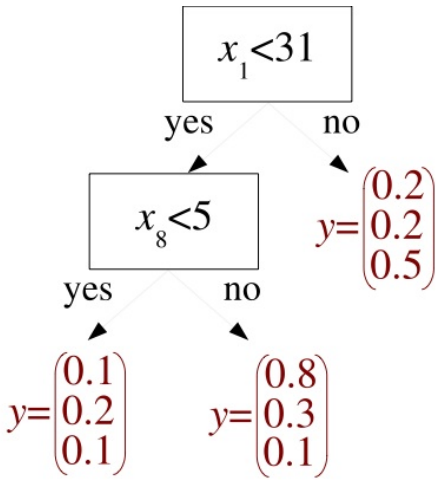
Outline

Regression trees with multiple outputs

Model h_{OK3}

$$h_{tree}(x(o)) = \frac{1}{n_L} \sum_{i=1}^{n_L} \bar{y}_i . 1_i(t(x(o)))$$

t is the tree indicator function that gives the number of the leaf where a data x falls. n_L the number of leaves $\bar{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j$



Regression trees in output feature space

- Given the current training set, the best split is the one that maximizes the empirical variance reduction:

$$\text{Score}_R(\mathit{Split}, S) = \text{var}\{y|S\} - \frac{n_l}{n}\text{var}\{y|S_l\} - \frac{n_r}{n}\text{var}\{y|S_r\},$$

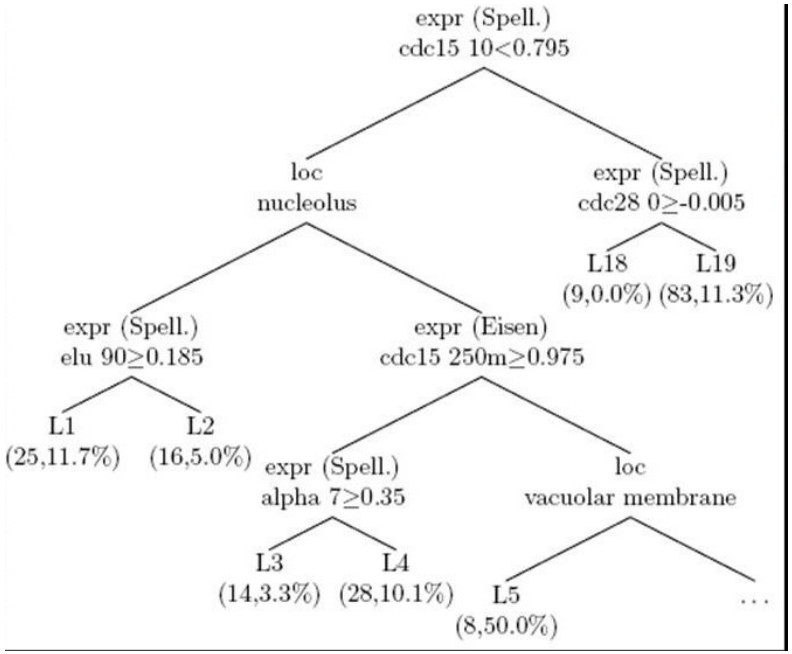
where n is the size of S, n_l (resp. n_r) the size of S_l (resp. S_r), and $\text{var}\{Y|S\}$ denotes the variance of the output Y in the subset S:

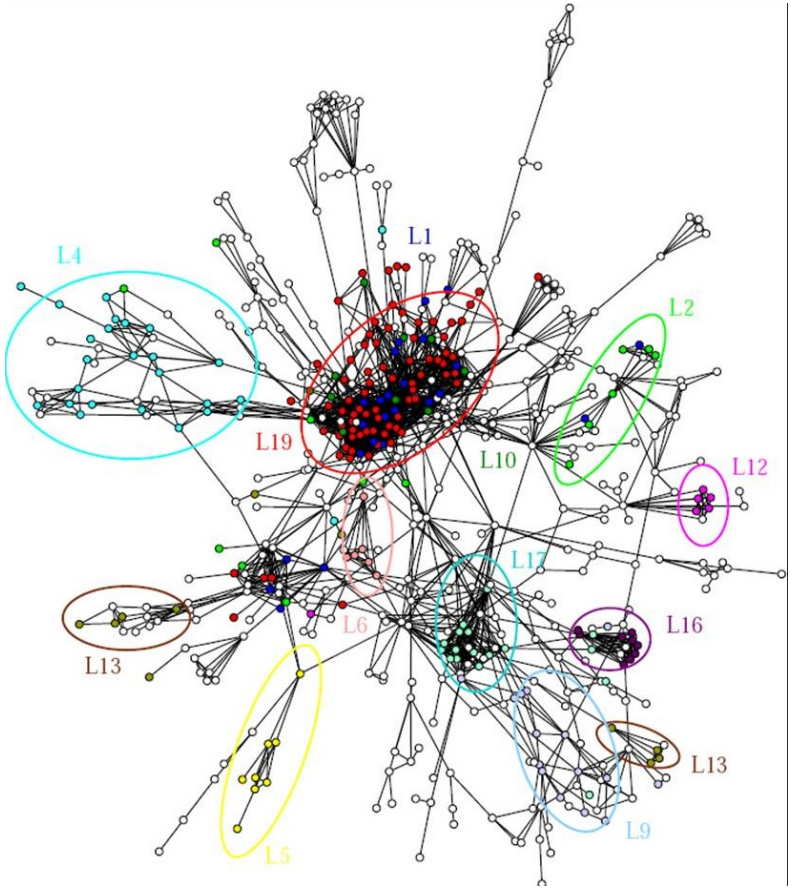
- $$\text{var}\{y|S\} = \frac{1}{n}\sum_{i=1}^{n} ||y_i - \overline{y}||^2 \text{ with } \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- Using kernel trick, we have:

$$\text{var}\{y|S\} = \frac{1}{n}\sum_{i=1}^{n} k_y(o_i, o_i) - \frac{1}{n^2}\sum_{i,j=1}^{n} k_y(o_i, o_j)$$

Example of a tree





[Geurts et al. 2007]

Application to yeast (*S. cerevisiae*)

- Protein-protein interaction network: 984 proteins, 2478 edges (Kato et al., 2005)
- Input features :
  - **Expression data:** expression of the gene in 157 experiments (Spellman's dataset and Eisen's dataset)
  - Phylogenetic profiles: presence or absence of an ortholog in 145 species
  - Localization data: presence or absence of the protein in 23 intracellular location
  - Yeast two hybrid data: data from a high-throughput experiment to detect protein-protein interactions
- Output features :
- graph of interactions

PPI yeast network results with 10 − *CV*

Protein network

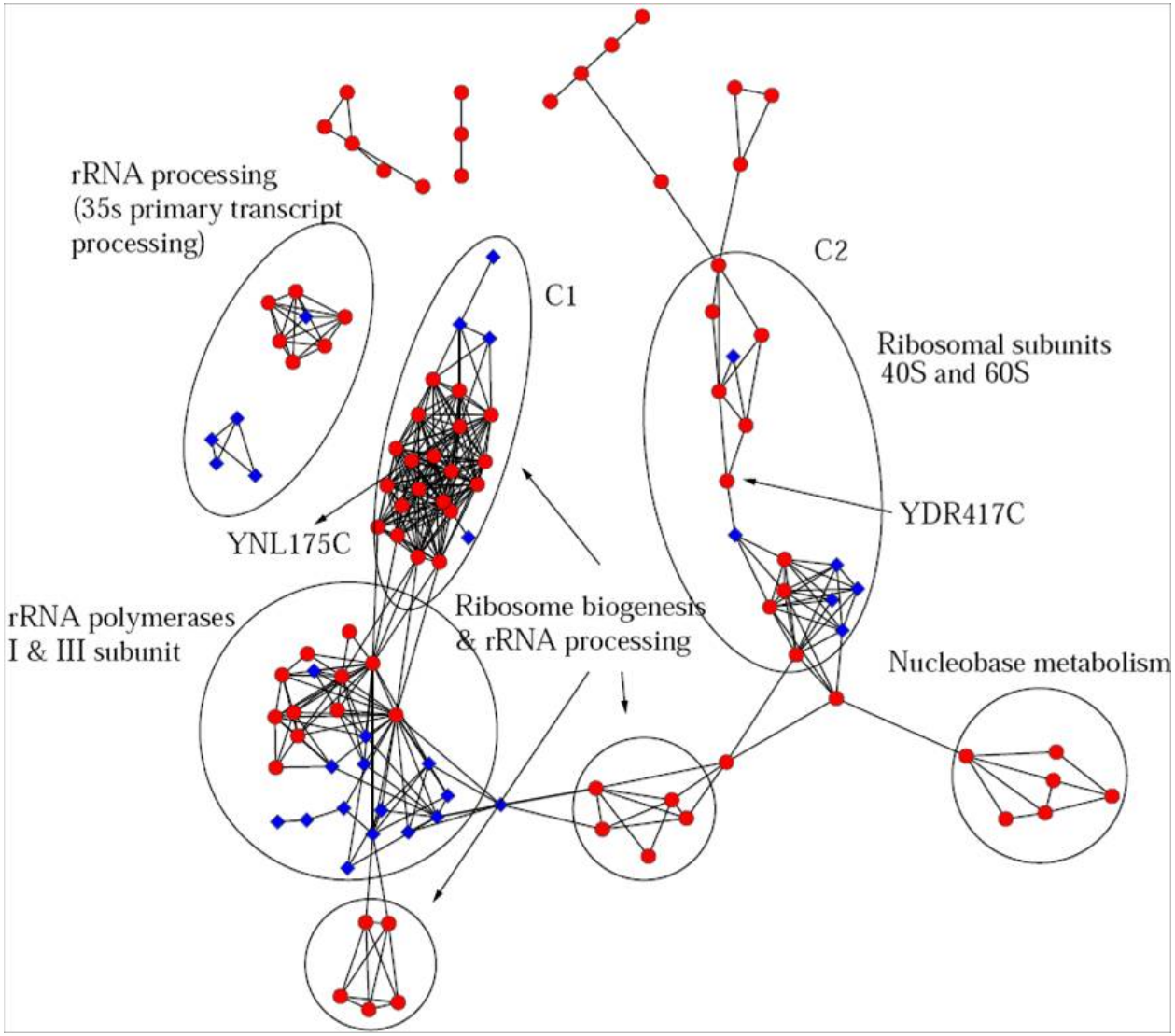
Inputs	OK3+ET	[1]
expr	0.851	0.776
phy	0.693	0.767
loc	0.725	0.788
y2h	0.790	0.612
All	**0.910**	**0.939**

Enzyme network

Inputs	OK3+ET	[2]
expr	0.714	0.706
phy	0.815	0.747
loc	0.587	0.577
All	**0.847**	**0.804**

- [1] Kato et al., ISMB 2005: EM based algorithm for kernel matrix completion
- [2] Yamanishi et al., ISMB 2005: compare a kernel canonical correlation analysis based solution and a metric learning approach
- *OK*3 : output kernel tree
- *ET* : extra-trees (specific random forests, proposed by Geurts et Wehenkel in 2004).

Function prediction for yeast data using ppi prediction



A model with input and output kernels

Additional assumption

- There exists a positive definite kernel k_x from $\mathcal{O} \times \mathcal{O}$ that codes for similaries between inputs

- Let 's call $x(\cdot)$ the feature map. We have: $k_x(o, o') = x(o)^t x(o')$
- Look for linear models from $\mathcal{X} \to \mathcal{Y}$

A model inspired from SVM and Maximum Margin Robot (Szedmak et al. 2005, Astikainen et al. 2009)

Let's define h_a such that $h_a(o) \in \mathcal{Y}$

$$h_a(x(o)) = \sum_{i=1}^{n} a_i y(o_i) x(o_i)^T x(o) \quad (4)$$

$$= \sum_{i=1}^{n} a_i y(o_i) k_x(o_i, o) \quad (5)$$

$= (Y_n I_a X_n^T) x(o)$(6)where $I_a = Diag(a)$ and Y_n (resp. X_n) is the matrix composed of the n vectors $y(o_i)$ (resp. $x(o_i)$),$i = 1...n$

Penalized least square loss with ℓ_2 norm

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

Closed form solution

$$\hat{a} = \left(K_{Y_n} \cdot K_{X_n} K_{X_n}^T + \lambda_1 I_n \right)^{-1} diag(K_{Y_n} K_{X_n}^T)$$

Proximity with Cortes et al.'s work on mapping strings to strings

- $K2R2 - A$: $h_A(x(o)) = Ax(o)$ [Cortès, Morhi,Weston ICML 2005]

Cost function

$$\min \sum_{i=1}^{n} \| h_A(o_i) - y(o_i) \|^2 + \lambda_1 \| A \|_F^2$$

Closed form solution

$$\hat{A} = Y_n \left(K_{X_n} + \lambda_1 I_n\right)^{-1} X_n^T \text{ (dual solution)}$$

Model use

- in Cortes et al.s' paper, this model was used to predict sequences so solving a pre-image problem was necessary
- Here, when predicting, we always directly use the dot product: $h_A(x(o))^T h_A(x(o'))$

A model richer that h_a less complex than h_A

- Even if only dot products are involved, we may be disturbed by the potentially infinite size of the matrix A ?
- Can we simplify the definition of h_A and complexify the definition of h_a which may be too simple
- Instead of diagonal matrix I_a in the model h_a, let us take an arbitrary matrix of size $n \times n$:
- A new model: $h_{A_n}(x(o)) = (Y_n A_n X_n^T)x(o)$

$$\min \sum_{i=1}^{n} \| h_{A_n}(o_i) - y(o_i) \|^2 + \lambda_1 \| (Y_n A_n X_n^T) \|_F^2$$

$$A_n = (K_{X_n} + \lambda_1 I_n)^{-1}$$

Comments

- In this supervised framework, h_A and h_{A_n} define the same function

PPI network: Yeast

Supervised setting with 10-CV, only gene expression

	OK3	OK3+ET	OK2$-a$	OK2*-A	[1]
AUC-ROC :	0.695	0.851	0.788	0.844	0.776

- [1] Kato et al., ISMB 2005: EM based algorithm for kernel matrix completion

Comparison with other works on PPI network

- Setting used in Bleakley et al. to compare network reconstruction performance with 10-CV
- Local methods only apply to predict interactions between learning set (LS) and test set (TS)
- Other methods can complete both LS and TS
- For our method:
  Hyperparameter selection performed by a 5-CV using AUF (area unc

Results for the h_a model on the ppi yeast network

AUROC

	exp	loc	phy	y2h	int
all	78.1 ± 1.5	66.5 ± 2.1	64.0 ± 1.3	50.4 ± 3.0	75.3 ± 2.2
ls vs ts	78.1 ± 1.2	66.5 ± 2.1	64.3 ± 1.4	51.1 ± 3.5	75.2 ± 1.7
ts vs ts	77.7 ± 4.7	66.7 ± 4.5	61.6 ± 1.6	46.9 ± 4.6	75.7 ± 5.9

AUF

	exp	loc	phy	y2h	int
all	93.1 ± 2.3	95.3 ± 0.7	99.1 ± 0.2	98.6 ± 0.4	93.7 ± 2.0
ls vs ts	93.2 ± 2.0	95.1 ± 0.6	99.1 ± 0.2	98.5 ± 0.5	93.7 ± 1.7
ts vs ts	91.8 ± 5.2	96.1 ± 1.4	99.1 ± 0.2	98.9 ± 1.0	93.6 ± 4.2

Results for the h_A model on the ppi yeast network

AUROC

	exp	loc	phy	y2h	int
all	82.8 ± 2.2	69.2 ± 1.8	69.0 ± 1.8	59.4 ± 3.3	90.4 ± 0.3
ts | ls	83.3 ± 2.1	69.2 ± 1.8	69.6 ± 1.5	60.8 ± 3.5	91.0 ± 0.4
ts | ts	78.4 ± 4.5	69.0 ± 4.5	64.7 ± 4.5	51.0 ± 5.5	86.1 ± 1.9

AUF

	exp	loc	phy	y2h	int
all	86.9 ± 4.4	95.3 ± 0.9	97.4 ± 0.3	88.1 ± 2.7	74.4 ± 6.2
ts | ls	86.3 ± 4.4	95.2 ± 0.8	97.4 ± 0.4	87.1 ± 2.9	72.8 ± 6.5
ts | ts	91.2 ± 4.0	95.6 ± 1.3	96.9 ± 0.8	93.8 ± 2.9	86.9 ± 5.0

Comparisons performed by Bleakley et al. 2007

Method	Metabolic				Protein–protein interactions				
	exp	loc	phy	int	exp	loc	phy	y2h	int
Direct	50.2 ± 1.5	56.5 ± 1.3	56.6 ± 0.7	57.6 ± 1.3	69.7 ± 0.5	67.0 ± 1.8	64.8 ± 2.6	50.2 ± 1.4	72.1 ± 1.1
kCCA	63.7 ± 1.6	58.2 ± 1.0	74.4 ± 1.7	74.7 ± 2.1	81.4 ± 1.1	49.1 ± 14.6	67.8 ± 2.2	48.1 ± 2.4	87.8 ± 0.9
kML	69.2 ± 2.6	57.0 ± 5.6	72.9 ± 0.8	76.9 ± 3.0	82.9 ± 1.2	76.3 ± 1.1	71.7 ± 1.8	64.4 ± 1.9	88.1 ± 1.2
em	66.4 ± 1.8	**65.6 ± 1.0**	**80.4 ± 1.5**	80.0 ± 1.8	80.6 ± 1.1	76.7 ± 3.8	71.0 ± 1.3	57.2 ± 2.7	**89.3 ± 1.1**
Pkernel	**70.3 ± 2.3**	58.2 ± 3.9	77.6 ± 2.3	80.0 ± 3.0	**83.8 ± 1.4**	**79.2 ± 2.6**	74.8 ± 1.8	67.5 ± 1.8	87.2 ± 0.8
local	69.3 ± 1.9	56.0 ± 3.3	67.8 ± 2.1	**82.1 ± 2.2**	78.1 ± 1.1	77.1 ± 2.9	**75.5 ± 2.4**	**77.8 ± 1.2**	87.6 ± 1.8

For each method ('Direct': direct method; 'kCCA': kernel canonical correlation analysis method; 'kML': kernel metric learning method; 'em': em projection method; 'Pkernel': tensor product pairwise kernel with SVM and 'local': local models with SVM), we show the AUC estimated by 5-fold cross-validation on the reconstruction of the metabolic gene network and the protein–protein interaction network from different datasets ('exp': expression data; 'loc': cellular localization; 'phy': phylogenetic profile and 'int': integration of the individual datasets).

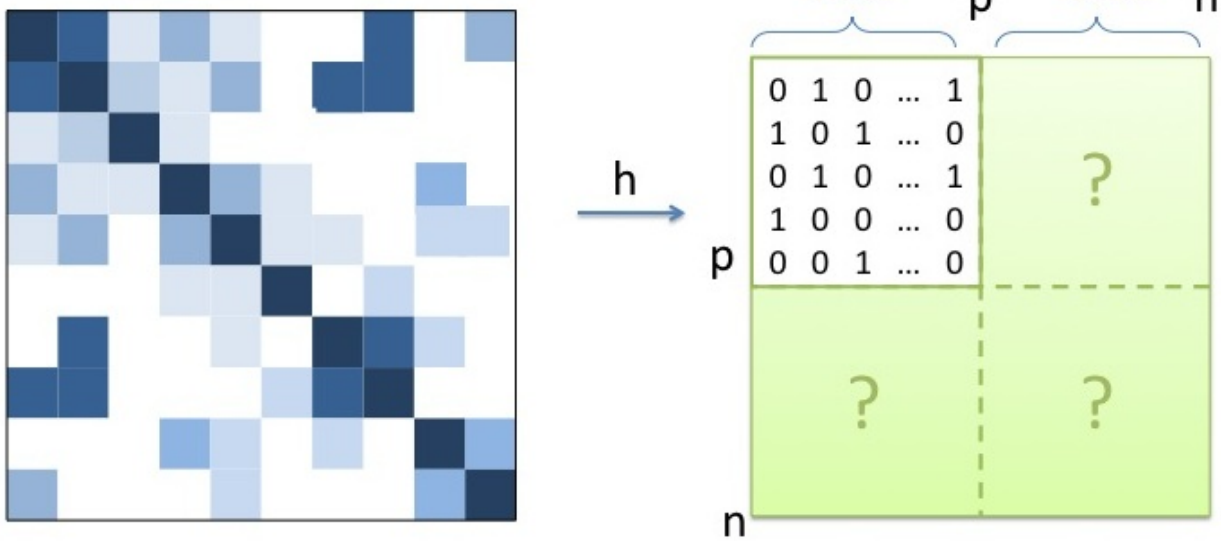
Table 2. Network reconstruction performance in terms of FDR

Method	Metabolic				Protein–protein interactions				
	exp	loc	phy	int	exp	loc	phy	y2h	int
Direct	98.7 ± 0.1	98.3 ± 0.2	98.3 ± 0.2	98.2 ± 0.2	98.4 ± 0.2	98.5 ± 0.2	98.5 ± 0.4	91.6 ± 1.2	94.3 ± 0.7
kCCA	96.9 ± 0.4	98.2 ± 0.1	89.4 ± 1.7	93.3 ± 0.8	**92.0 ± 1.9**	97.9 ± 3.8	98.1 ± 0.4	97.9 ± 0.6	88.6 ± 1.5
kML	97.2 ± 0.5	98.3 ± 0.2	95.1 ± 0.9	94.0 ± 1.2	93.0 ± 2.4	98.8 ± 0.1	98.5 ± 0.2	99.2 ± 0.1	93.5 ± 2.2
em	97.0 ± 0.1	**97.2 ± 0.1**	**84.2 ± 2.3**	89.0 ± 0.7	93.7 ± 1.2	**94.5 ± 1.1**	**96.8 ± 0.5**	**89.6 ± 1.0**	80.9 ± 1.3
Pkernel	95.5 ± 0.2	98.3 ± 0.2	93.1 ± 0.7	89.5 ± 2.3	92.4 ± 1.0	95.1 ± 1.0	98.2 ± 0.3	98.5 ± 0.5	88.6 ± 2.2
local	**93.9 ± 1.9**	98.2 ± 0.5	96.1 ± 0.4	**59.4 ± 2.4**	97.4 ± 0.4	96.3 ± 0.9	97.9 ± 0.3	92.4 ± 1.6	**74.5 ± 3.4**

This table shows the performance of each method on each benchmark experiment in terms of AUF.

Outline

Matrix completion or supervised task ? A transductive task



Towards a more realistic task

- When building a classifier to predict interactions between proteins, all the proteins are available during the training phase
- Let us use the input features of all the proteins
- The learning task is a transductive one: we do not want to build a general predictor that works for any protein dataset, but for the set at hand.
- Here we will still build a function but will only evaluate it on the data at hand
- We will assume a very small number of known edges

Semi-supervised Output Kernel regression

- Impose a smoothness assumption on function h with regularization operators based on Laplacian (Zhu et al. 2003, Zhou et al. 2004, Belkin et al. 2005, Lafferty and Wasserman, 2007)
- Need a smooth model:
  - Trees are not appropriate (while linear combination could be eligible)
  - Input output kernel regression with penalized least square is eligible

Regularized least square loss for semi-supervised learning

Notations (be careful): p = labeled examples and n = total number of examples

$$\min \sum_{i=1}^{p} \| h_a(x_i) - y(o_i) \|^2 + \lambda_1 \| a \|^2 + \lambda_2 \sum_{j=1}^{n} \sum_{j'=1}^{n} k_x(o_j, o_{j'}) \| h_a(x_j) - h_a(x_{j'}) \|^2$$

with:

$$\sum_{j=1}^{n} \sum_{j'=1}^{n} k_x(o_j, o_{j'}) \| h_a(x_j) - h_a(x_{j'}) \|^2 = tr(h_a L h_a^t) \quad (7)$$

- $L = D_x - K_x$ with D_x the diagonal matrix such that $d_x(i,i) = \sum_j k_x(i,j)$

Closed form solution for h_a

Closed form solution for h_a

$$a = \left(K_{Y_p} \cdot K_{X_p} K_{X_p}^t + \lambda_1 I_p + 2\lambda_2 K_{Y_p} \cdot K_{X_{pn}} L K_{X_{np}} \right)^{-1} diag(K_{Y_p} K_{X_p}^t)$$

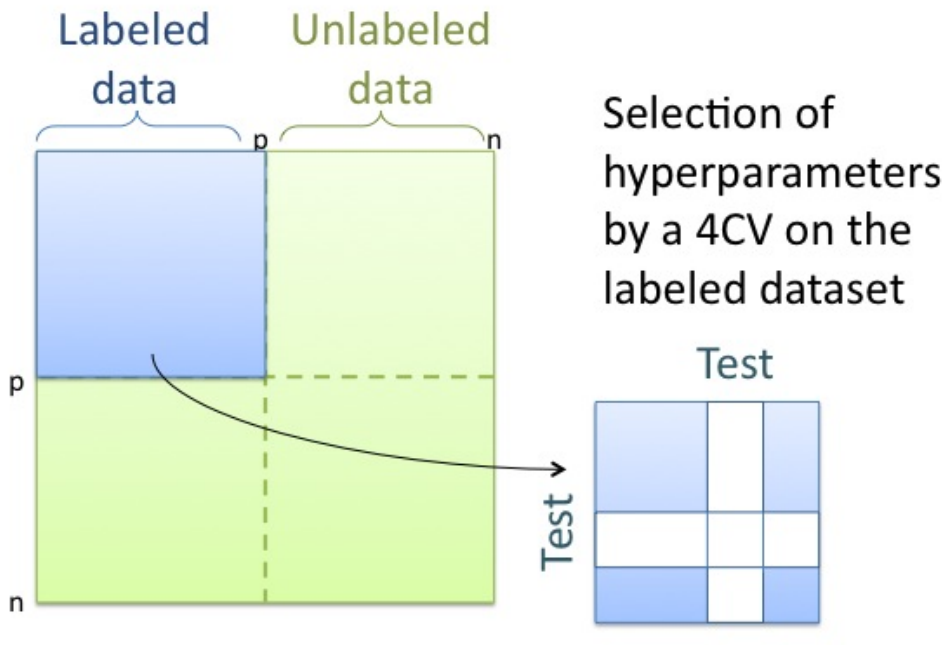
Closed form solution for h_A and h_{A_n}

$$A = Y_p \left(K_{X_p} + \lambda_1 I_p + 2\lambda_2 K_X L \right)^{-1} X^t \text{ (dual solution)}$$

$$A_n = \left(K_{X_p} + \lambda_1 I_p + 2\lambda_2 K_{X_{pn}} L K_{X_{np}} K_{X_p}^{-1} \right)^{-1}$$

Yeast PPI: protocol for hyperparameter selection

We drew $B = 10$ samples of labeled data of size p.



Yeast PPI: results for $K2R2 - a$

Average results (AUROC) on 10 samples

Only gene expression

	p = 0.1 n	p = 0.2 n	p = 0.5 n
Supervised	0.685 ± 0.082	0.755 ± 0.015	0.778 ± 0.012
Transductive _ Kdiff	0.740 ± 0.032	0.772 ± 0.019	0.793 ± 0.012

- Notice that in previous supervised results with $10-$CV: 90% of training examples were used in each fold
- Here, $p = 10\%n$ of nodes means taking roughly 1% of the edges in the labeled training set
- Other transductive methods are disadvantaged by our setting because we assume to know a small submatrix and not randomly sampled coefficients in the matrix

Conclusion

- **Take-home message**: use the kernel trick in the output space ! spend time on the choice of output space !
- As kernel ridge regression, Input Output Kernel Regression with penalized least square keeps closed form solution for a lot of constraints
- First promising results on the transductive task: improvement obtained when using unlabeled data
- Realistic tasks using a very few labeled examples compared to literature

Perspective

- Application to human proteome (network of CFTR whose mutations are responsible for cystic fibrosis)
- Dealing with unbalanced set of examples (absence of negatives)
- Integration and automated selection of multiple input kernels
- Encapsulate the model into a graphical probabilistic one (logistic regression-like)
- Many other problems of structure prediction can be tackled with this framework...if you find the right output space !

Acknowledgements

- IBISC - Université d'Evry, Genopole, France
  - Céline Brouard (PhD student)
  - Marie Szafranski
- GIGA-R, Université de Liège, Belgium
  - Pierre Geurts
  - Louis Wehenkel
- Radboud University Nijmegen, The Netherlands
  - Adriana Birlutiu
  - Tom Heskes
- Hopital Cochin, INSERM, Paris, France
  - Aleksander Edelman
  - Chiara Guerrera