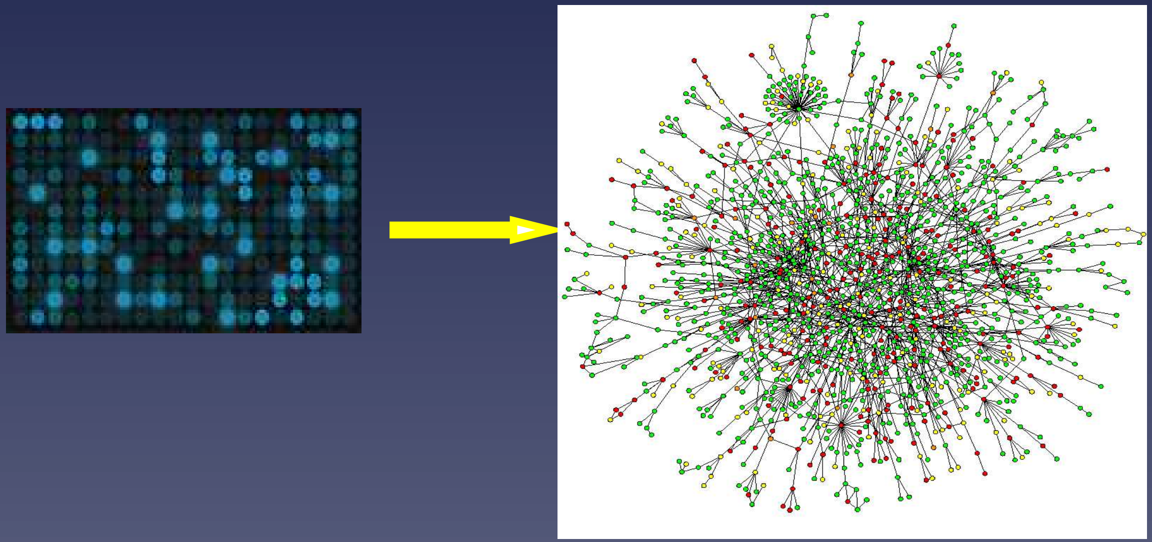
Machine learning approches for reconstruction of genetic networks

Jean-Philippe Vert
Ecole des Mines de Paris
Center for Computational Biology
Jean-Philippe.Vert@mines.org

Workshop on Knowledge Discovery and Emergent Complexity in BioInformatics (KDECB 2006), Ghent, May 10th, 2006

Motivations: systems biology



- Gene expression
- Sequence
- Protein structure
- Protein localization, etc...

- Regulatory network
- Signaling pathways
- Metabolic pathways
- Interaction network, etc...

Mains approaches

1. Direct approach = connect *similar* proteins.

2. Model-based approach = fit an *a priori* defined model (Bayesian network, dynamical system..).

3. Indirect approach = connect pairs of proteins *similar* to connected pairs.

Machine learning is present in all 3 approaches.

Indirect approach

- Classical setting of supervised pattern recognition: "given a training set of connected and non-connected pairs, learn to predict whether new pairs are connected or not".
- Need to extend the representation of points to the representation of pairs of points.
- Example: a pairwise kernel (Ben-Hur and Noble, 2004):

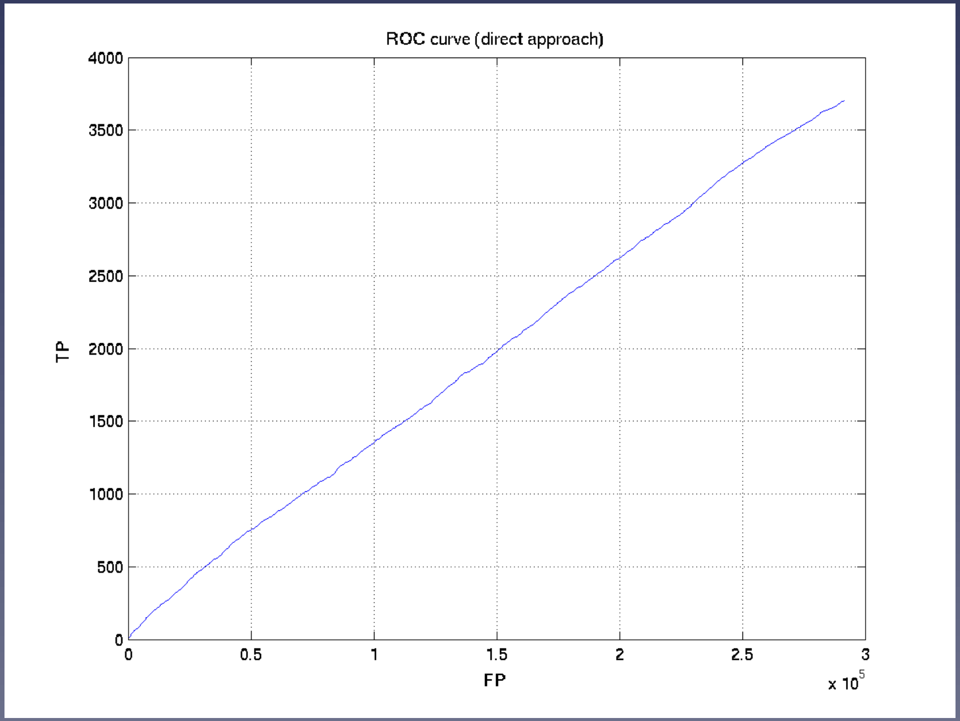
$$K_p\left((u_1, u_2), (v_1, v_2)\right) = K(u_1, v_1)K(u_2, v_2) + K(u_1, v_2)K(u_2, v_1)$$

Direct approach

- The simplest and most natural approach.
- Define a measure of similarity (e.g., correlation coefficient between expression profiles) and connect the most similar pairs.
- Usually unsupervised, but..

Performance of unsupervised direct approach

The metabolic network of the yeast involves 769 genes. Each gene is represented by 157 expression measurements. (ROC=0.52)



What is wrong?

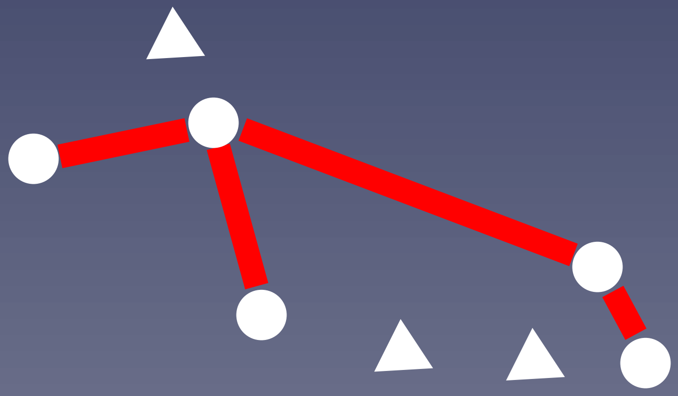
- What similarity measure between profiles should be use?

What is wrong?

- What similarity measure between profiles should be use?
- Which network are we expecting to recover?

Supervised direct approach

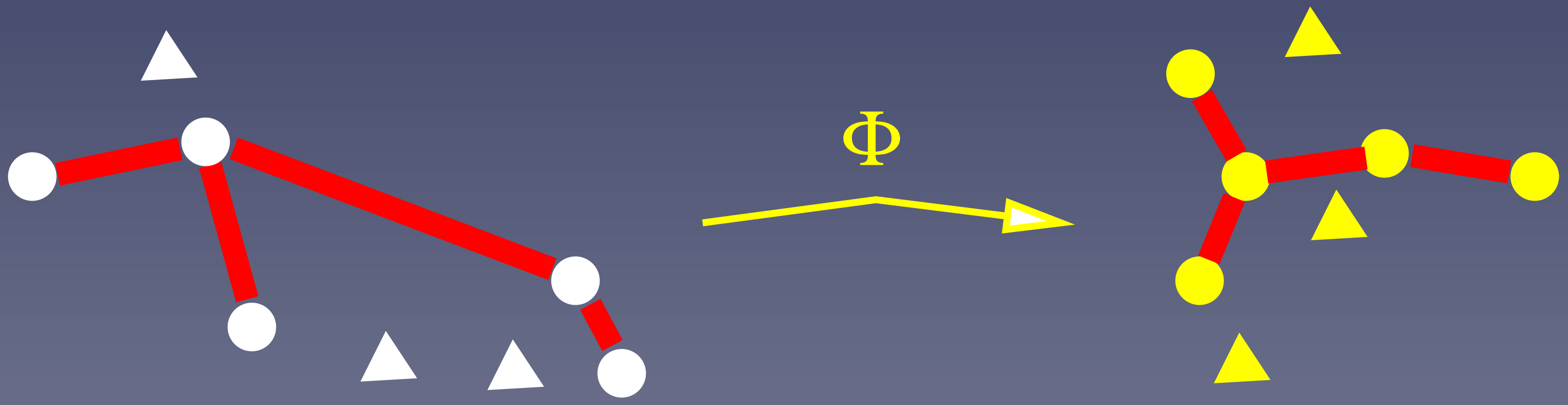
- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs
- Typical problem of distance metric learning



Supervised direct approach

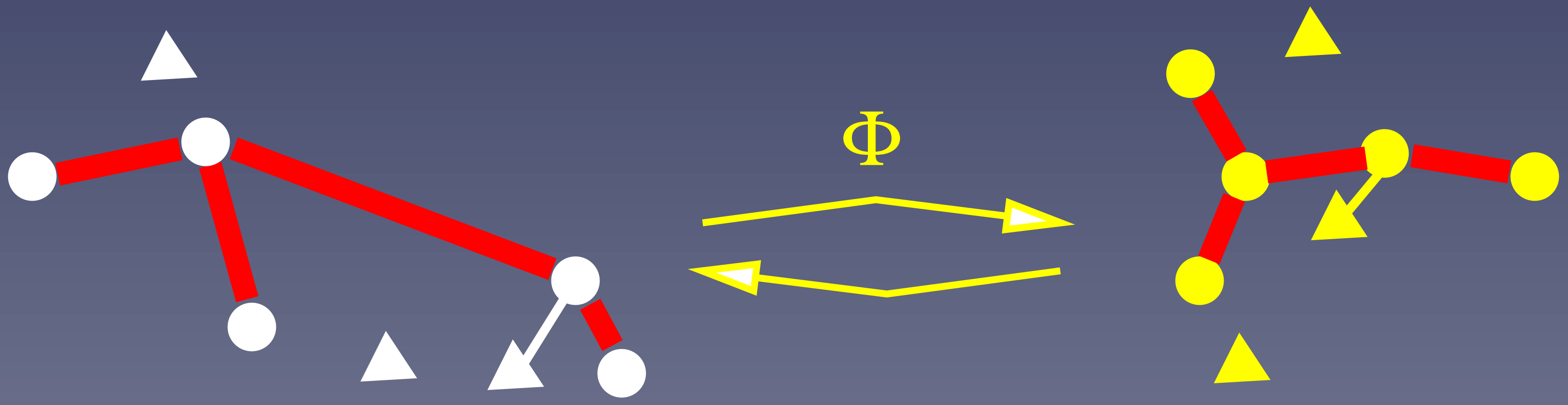
- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs

- Typical problem of distance metric learning



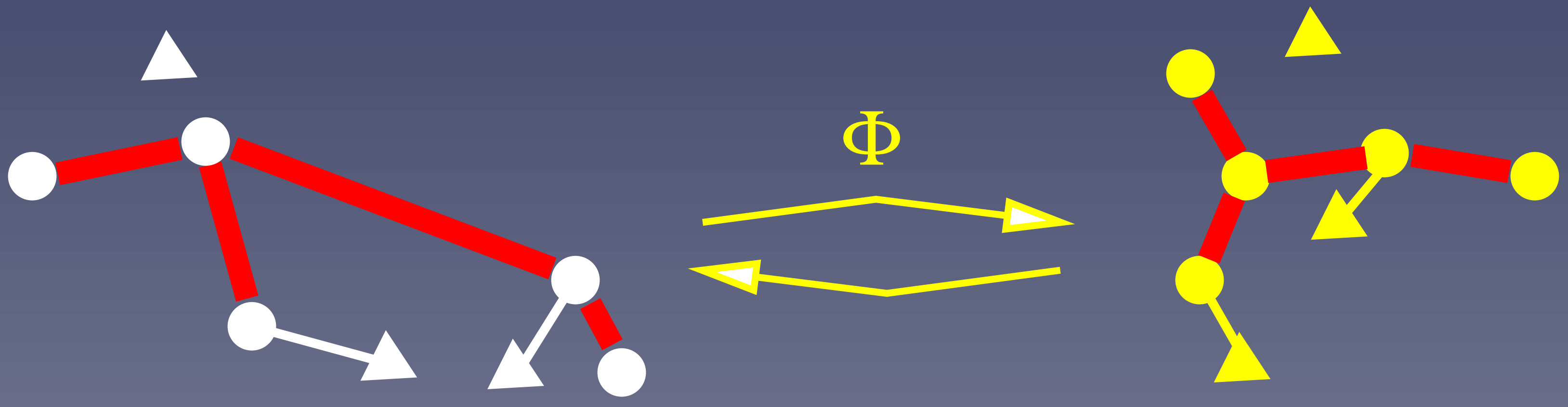
Supervised direct approach

- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs
- Typical problem of distance metric learning



Supervised direct approach

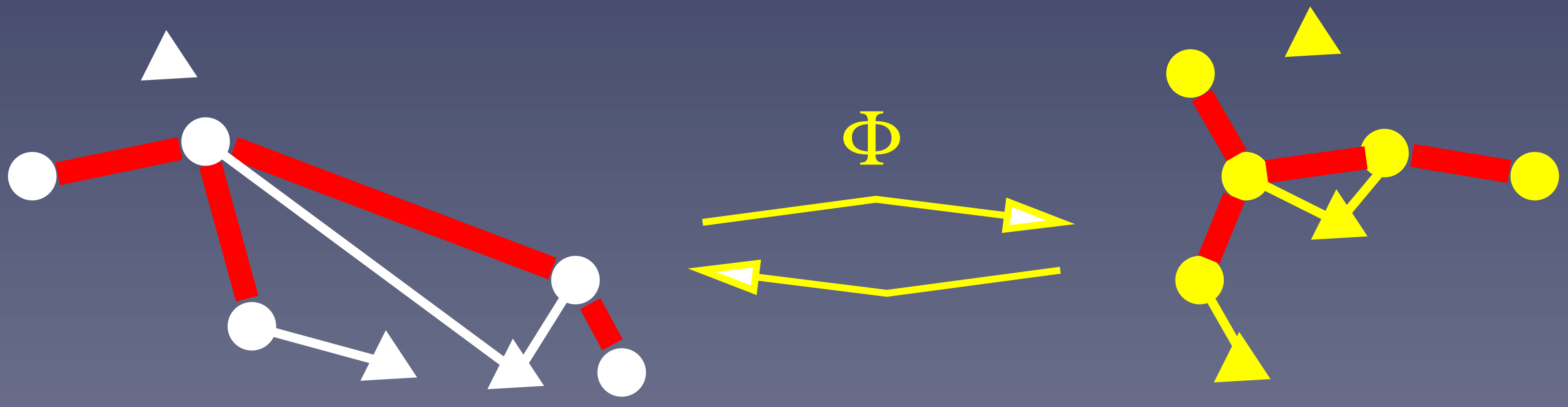
- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs
- Typical problem of distance metric learning



Supervised direct approach

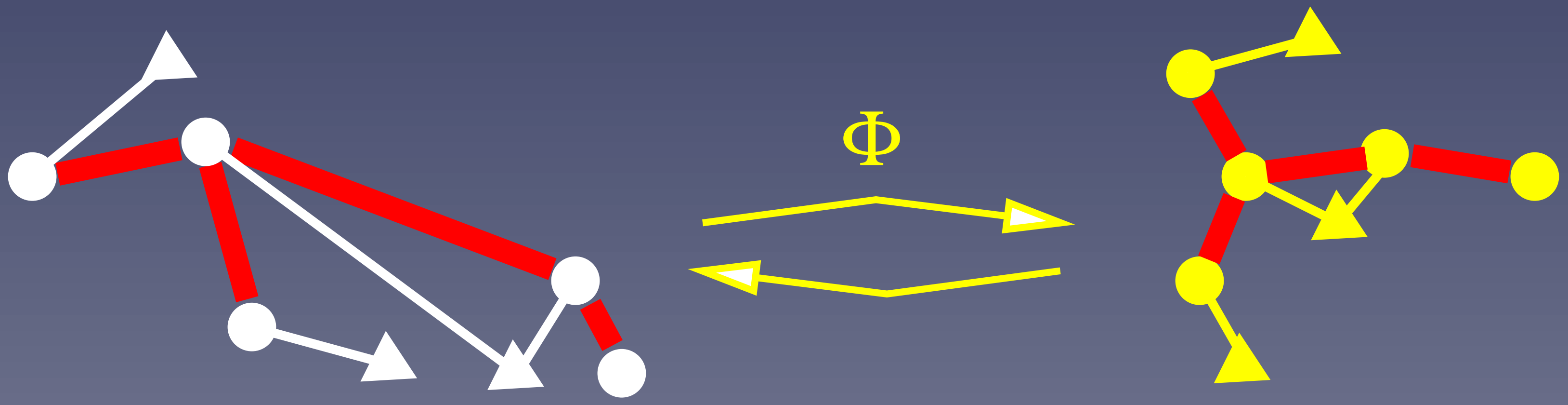
- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs

- Typical problem of distance metric learning



Supervised direct approach

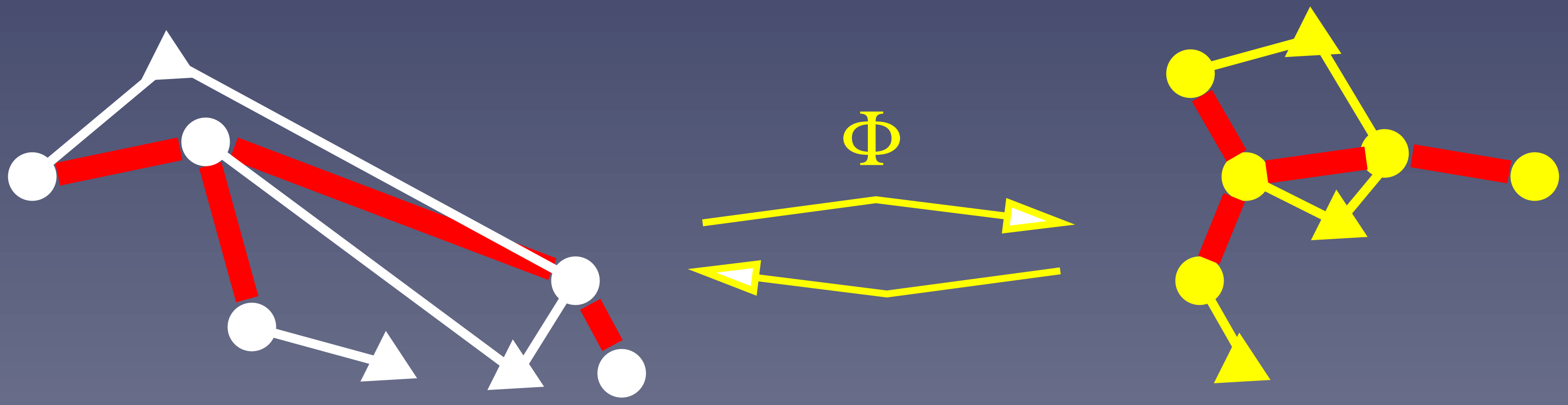
- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs
- Typical problem of distance metric learning



Supervised direct approach

- Given a set of known interacting pairs, we can learn how to measure their similarities before connecting similar pairs

- Typical problem of distance metric learning



Part 2

Supervised direct inference by generalized KPCA

Explicit mapping Φ

- Let $x \in \mathbb{R}^p$ be a genomic data (e.g., expression profile)

Explicit mapping Φ

- Let $x \in \mathbb{R}^p$ be a genomic data (e.g., expression profile)
- Let us consider linear mappings:

$$\Phi(x) = (f_1(x), \ldots, f_d(x))' \in \mathbb{R}^d$$

made of linear features $f_i(x) = w_i^\top x$

Explicit mapping Φ

- Let $x \in \mathbb{R}^p$ be a genomic data (e.g., expression profile)
- Let us consider linear mappings:

$$\Phi(x) = (f_1(x), \dots, f_d(x))' \in \mathbb{R}^d$$

made of linear features $f_i(x) = w_i^\top x$

- A feature $f : \mathbb{R}^p \to \mathbb{R}$ is “good” if connected genes in the known network have similar value.

“Good” features

- A “good” feature $f(x) = w^\top x$ should minimize:

$$R(f) = \frac{\sum_{i \sim j} \left(f(x_i) - f(x_j)\right)^2 - \sum_{i \not\sim j} \left(f(x_i) - f(x_j)\right)^2}{\sum_{i=1}^{n} f(x_i)^2}$$

“Good” features

- A “good” feature $f(x) = w^\top x$ should minimize:

$$R(f) = \frac{\sum_{i\sim j} \left(f(x_i) - f(x_j)\right)^2 - \sum_{i\not\sim j} \left(f(x_i) - f(x_j)\right)^2}{\sum_{i=1}^n f(x_i)^2}$$

- Regularisation: for statistical reasons, it is safer to minimize:

$$\min_{f(x)=w^\top x} R(f) + \lambda \frac{\|w\|^2}{\sum_{i=1}^n f(x_i)^2}$$

Influence of λ

- $\lambda \to +\infty$: PCA
  - ⋆ Useful for noisy, high-dimensional data.
  - ⋆ Used in spectral clustering. The graph does not play any role (unsupervised)
- $\lambda \to 0$: second smallest eigenvector of the graph
  - ⋆ Useful to embed the graph in a Euclidean space (used in graph partitioning)
  - ⋆ Sensitive to noise. Mapping of points outside of the graph unstable (overfitting)

Extracting successive features

- Successive features to form Φ can be obtained by:

$$w_i = \arg\min_{w \perp \{w_1, \ldots, w_{i-1}\}, \hat{\text{var}}(f_w)=1} \left\{ \sum_{i \sim j} \left(f_w(x_i) - f_w(x_j)\right)^2 + \lambda \|w\|^2 \right\}.$$

Extracting successive features

- Successive features to form Φ can be obtained by:

$$w_i = \mathop{\arg\min}_{w \perp \{w_1,\ldots,w_{i-1}\},\, \hat{\text{var}}(f_w)=1} \left\{ \sum_{i\sim j} \left(f_w(x_i) - f_w(x_j)\right)^2 + \lambda \|w\|^2 \right\}.$$

- Generalizes Principal Component Analysis (PCA)

Limitations

- How to generalize to non-linear features?
- How to process non-vectorial data (sequences, phylogenetic profiles, ...)

Overcoming the limitations

- Remember:

$$w_i = \operatorname*{arg\,min}_{w \perp \{w_1,\ldots,w_{i-1}\},\, \hat{\text{var}}(f_w)=1} \left\{ \sum_{i \sim j} (f_w(x_i) - f_w(x_j))^2 + \lambda \|w\|^2 \right\}.$$

- In order to allow nonlinear features, we need to replace:

  $\star$ $\|w\|^2$ by $\|f\|^2$

  $\star$ $w_i \perp w_j$ by $f_i \perp f_j$

Positive definite kernels

Let $\mathcal{X}$ be a set (not necessarily vectors) endowed with a symmetric measure of similarity $k : \mathcal{X}^2 \to \mathbb{R}$ that satisfies:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

for any $n \geq 0, (x_1, \ldots, x_n) \in \mathcal{X}$ and $(a_1, \ldots, a_n) \in \mathbb{R}$

- $k(x, y) = x \cdot y$ for $\mathcal{X} = \mathbb{R}^d$
- $k(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$ for $\mathcal{X} = \mathbb{R}^d$

Reproducing kernel Hilbert space

- A p.d. kernel defines a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ obtained by completing the span of $\{k(x,\cdot), x \in \mathcal{X}\}$
- The norm of a function $f(x) = \sum_{i=1}^{n} c_i k(x_i, x)$ is:

$$\|f\|_k^2 = \sum_{i,j=1}^{n} c_i c_j k(x_i, x_j).$$

- This space is called the reproducing kernel Hilbert space (RKHS)

Example: linear RKHS

For $\mathcal{X} = \mathbb{R}^d$ and $k(x,y) = x \cdot y$, we have:

- $f(x) = \sum_{i=1}^{n} c_i x_i \cdot x = f_w(x)$ with $w = \sum_{i=1}^{n} c_i x_i$.
- $\| f \|_k^2 = \sum_{i,j=1}^{n} c_i c_j x_i \cdot x_j = \| w \|^2$
- If $f(x) = w \cdot x$ and $g(x) = v \cdot x$ then:

$$< f, g >_k = w \cdot v$$

Graph-driven feature extraction in RKHS

- For a general set $\mathcal{X}$ endowed with a p.d. kernel k we therefore have the following graph-driven feature extractor:

$$f_i = \operatorname*{arg\,min}_{f \perp \{f_1,\ldots,f_{i-1}\}, \hat{\mathrm{var}}(f)=1} \left\{ \sum_{i \sim j} \left(f(x_i) - f(x_j)\right)^2 + \lambda \| f \|_k^2 \right\}.$$

- The values at the minima (the spectrum) quantifies how much the graph fits the data

Solving the problem

- By the representer theorem, f_i can be expanded as:

$$f_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_i, x).$$

- This shows that

$$\begin{aligned} < f_i, f_j >_k &= \alpha_i^\top K \alpha_j \\ \|f_i\|_k^2 &= \alpha_i^\top K \alpha_i \end{aligned} \quad (1)$$

Solving the problem (cont.)

- The problem can then be rewritten:

$$\alpha_i = \underset{\alpha\in\mathbb{R}^n, \alpha K_V \alpha_1 = \ldots = \alpha K_V \alpha_{i-1} = 0}{\arg\min} \left\{ \frac{\alpha^\top K_V L K_V \alpha + \lambda \alpha^\top K_V \alpha}{\alpha^\top K_V^2 \alpha} \right\}$$

 where K_V is the centered $n \times n$ Gram matrix and L is the Laplacian of the graph

- It is equivalent to solving the generalized eigenvalue problem:

$$(LK_V + \lambda I)\alpha = \mu K_V \alpha.$$

Kernels

Several similarity kernels have been developed recently:

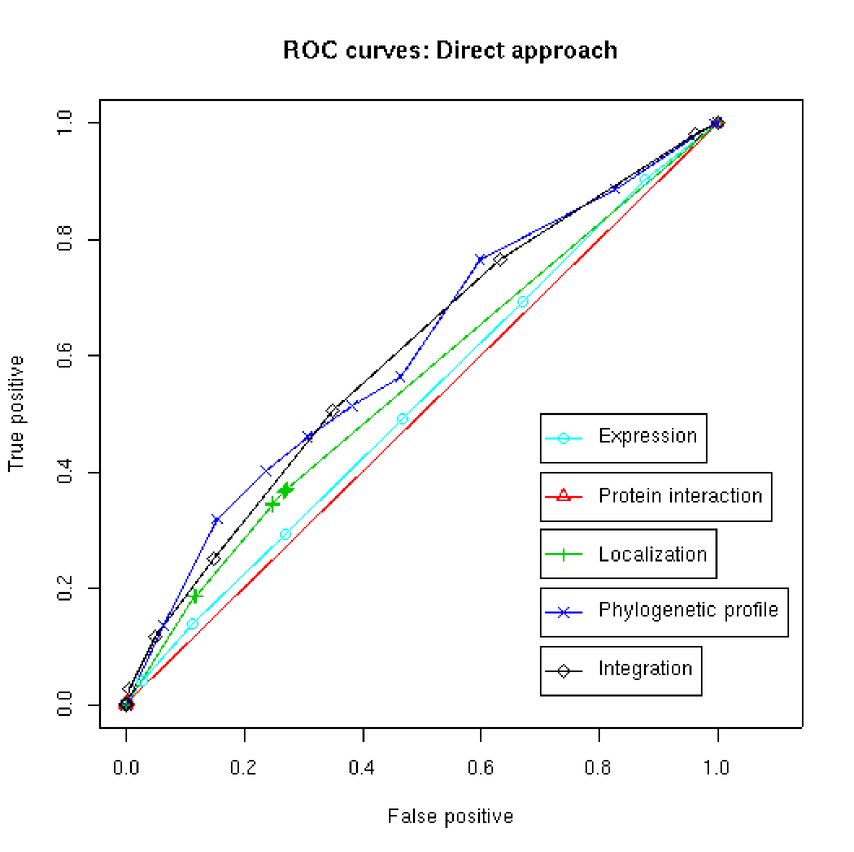
- for phylogenetic profiles (JPV. 2004)
- for gene sequences (Leslie et al. 2003, Saigo et al. 2004, ...)
- for nodes in a network (Kondor et al. 2000)

Learning from heterogeneous data

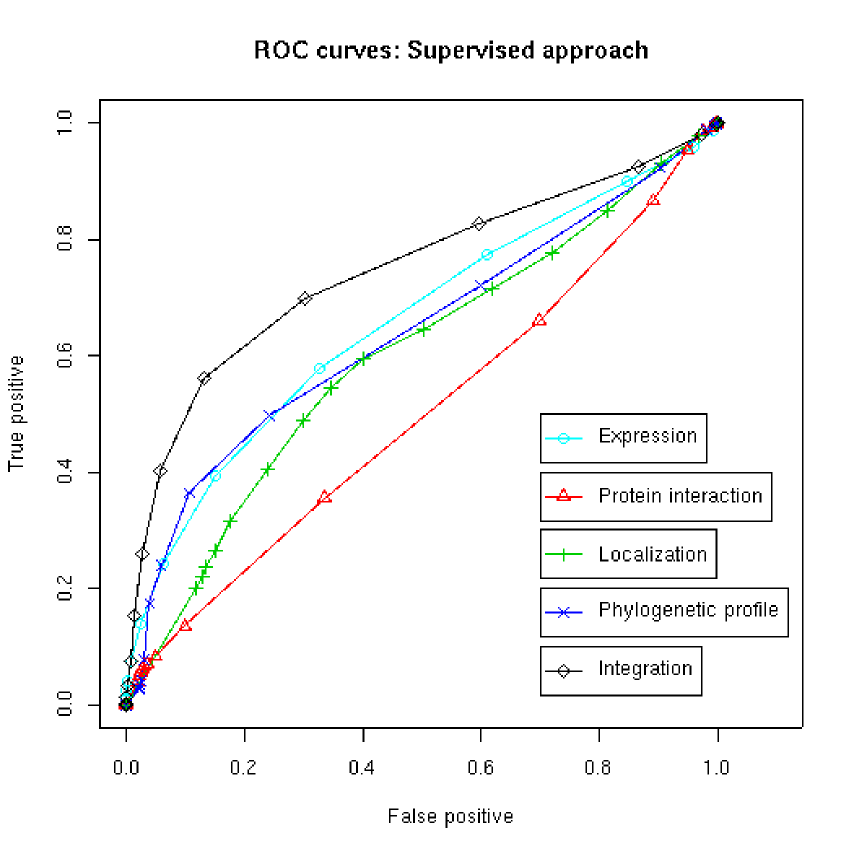
- Suppose several data are available about the genes, e.g., expression, localization, struture, predicted interaction etc...
- Each data can be represented by a positive definite similarity matrix $K_1, \ldots, K_p$
- Kernel can be combined by various operations, e.g., addition:

$$K = \sum_{i=1}^{p} K_i$$

Learning from heterogeneous data (unsupervised)



Learning from heterogeneous data (supervised)



Part 3

Supervised direct inference by metric learning pairwise kernel

Limitations of GKPCA

- Requires the training set to be made of the presence / absence of edges among a particular subset of genes
- Discrepancy between the objective function and the goal of edge inference
- Requires the tuning of two regularization parameters (d and λ)

Objective function

After a linear mapping $\Phi(x) = Ax$ the square Euclidean distance is:

$$\begin{aligned} d_M(x, x') &= (x - x')^\top M (x - x') \\ &= tr\left(M(x - x')(x - x')^\top\right) \ , \end{aligned}$$

with $M = A^\top A \succ 0$. Direct edge inference is possible if, for example,

$$d_\phi(x_i, x_j) \begin{cases} \leq \gamma - 1 & \text{for } x_i \sim x_j \ , \\ \geq \gamma + 1 & \text{for } x_i \not\sim x_j \ . \end{cases}$$

Large-margin metric learning

In the spirit of SVM, this suggests the following optimization problem:

$$\begin{aligned}
\text{Minimize} \quad & \| M \|_{Fro}^2 + C \sum_{(i,j)} \zeta_{i,j} \\
\text{subject to} \quad & \zeta_{i,j} \geq 0 \,, \quad \forall (i,j) \\
& d_M(x_i, x_j) \leq \gamma - 1 + \zeta_{i,j} \,, \quad i \sim j \\
& d_M(x_i, x_j) \geq \gamma + 1 - \zeta_{i,j} \,, \quad i \not\sim j \\
& M \succ 0 \,.
\end{aligned}$$

SVM formulation

If we relax the constraint $M \succ 0$ this is equivalent to a SVM:

$$\begin{aligned} \text{Minimize} \quad & \| M \|^2_{Fro} + C \sum_{(i,j)} \zeta_{i,j} \\ \text{subject to} \quad & \zeta_{i,j} \geq 0 \ , \quad \forall (i,j) \\ & < M, D_{i,j} >_{Fro} - \gamma \leq -1 + \zeta_{i,j} \ , \quad i \sim j \\ & < M, D_{i,j} >_{Fro} - \gamma \geq 1 - \zeta_{i,j} \ , \quad i \not\sim j \ . \end{aligned}$$

Inner product for pairs

The inner product between two pairs for this SVM is:

$$
\begin{aligned}
& K_p\left(\left(x_1, x_2\right),\left(x_3, x_4\right)\right) \\
& =\left\langle D_{x_1, x_2}, D_{x_3, x_4}\right\rangle_{Fro} \\
& =Trace\left(\left(x_1-x_2\right)\left(x_1-x_2\right)^{\top}\left(x_3-x_4\right)\left(x_3-x_4\right)^{\top}\right) \\
& =\left(\left(x_1-x_2\right)^{\top}\left(x_3-x_4\right)\right)^2 \\
& =\left(x_1^{\top} x_3-x_1^{\top} x_4-x_2^{\top} x_3+x_2^{\top} x_4\right)^2 .
\end{aligned}
$$

Metric learning pairwise kernel

If we start from a kernel K_g between single genes, this formulation is therefore a SVM to discriminate between connected and non-connected pairs with the following pairwise kernel:

$$K_{MLPK}\left(\left(x_1, x_2\right), \left(x_3, x_4\right)\right) \\ = \left(K_g\left(x_1, x_3\right) - K_g\left(x_1, x_4\right) - K_g\left(x_2, x_3\right) + K_g\left(x_2, x_4\right)\right)^2 .$$

To be compared, e.g., with the pairwise kernel:

$$K_p\left((x_1, x_2), (x_3, x_4)\right) = K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3) .$$

Experimental results

Prediction of the co-complex protein network for the yeast from various protein data (AUC performance in cross-validation)

Data	K_p	K_{MLPK}
Co-regulation (Chip-chip)	0.68	0.90
Co-localization	0.83	0.78
PFAM kernel	0.92	0.98
PSI-BLAST kernel	0.94	0.97

Conclusion

Conclusion

1. Supervised inference is better than unsupervised

Conclusion

1. Supervised inference is better than unsupervised

2. Supervised graph inference can be performed by distance metric learning

Conclusion

1. Supervised inference is better than unsupervised
2. Supervised graph inference can be performed by distance metric learning
3. Different formulations lead to different algorithms. New pairwise kernel.

Conclusion

1. Supervised inference is better than unsupervised
2. Supervised graph inference can be performed by distance metric learning
3. Different formulations lead to different algorithms. New pairwise kernel.
4. Data integration with kernels is simple and powerful

Conclusion

1. Supervised inference is better than unsupervised
2. Supervised graph inference can be performed by distance metric learning
3. Different formulations lead to different algorithms. New pairwise kernel.
4. Data integration with kernels is simple and powerful
5. Few assumptions about the network to infer (works well for the metabolic network and the protein interaction network)

Thanks

- Yoshihiro Yamanishi (Kyodai) : generalized KPCA
- Bill Noble, Jian Qiu (UW) : MLPK