Supervised inference of biological networks

Jean-Philippe Vert
Jean-Philippe.Vert@ensmp.fr

Centre for Computational Biology
Ecole des Mines de Paris, ParisTech

Machine Learning in Systems Biology (MLSB 2007), Evry, France,
September 24th, 2007
Outline

1 Motivation

2 Unsupervised inference

3 Supervised inference
   • Metric learning
   • Matrix completion
   • Global pattern recognition
   • Local pattern recognition

4 Experiments

5 Conclusion
Outline

1 Motivation

2 Unsupervised inference

3 Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4 Experiments

5 Conclusion
Outline

1. Motivation
2. Unsupervised inference
3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition
4. Experiments
5. Conclusion
Outline

1 Motivation

2 Unsupervised inference

3 Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4 Experiments

5 Conclusion
Outline

1. Motivation

2. Unsupervised inference

3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4. Experiments

5. Conclusion
Many interesting biological situations can be represented as network:
- Protein-protein interactions,
- Metabolic pathways,
- Signaling pathways, ...
Example: metabolic network

Vertices are enzymes

Edges connect two enzymes when they catalyze two successive reactions
What are the challenges?

Questions

1. Given a newly discovered protein (e.g. from genome sequencing), predict which known ones are connected to it.
2. Discover new functional relationships (new edges) between already known proteins.

Applications

- Genome annotation
- Elucidation of new pathways
- Prediction of new binding partners
- Identification of new candidate drug targets
What are the challenges?

Questions

1. Given a newly discovered protein (e.g. from genome sequencing), predict which known ones are connected to it.
2. Discover new functional relationships (new edges) between already known proteins.

Applications

- Genome annotation
- Elucidation of new pathways
- Prediction of new binding partners
- Identification of new candidate drug targets
**How can bioinformatics help?**

*Biologists* have collected a lot of data about proteins, e.g.,

- Gene expression measurements
- Phylogenetic profiles
- Location of proteins/enzymes in the cell

How to use this information "intelligently" to find a good function that predicts edges between nodes.
Our goal: Summary

Data
- Gene expression,
- Gene sequence,
- Protein localization, ...

Graph
- Protein-protein interactions,
- Metabolic pathways,
- Signaling pathways, ...
Outline

1 Motivation

2 Unsupervised inference

3 Supervised inference
   • Metric learning
   • Matrix completion
   • Global pattern recognition
   • Local pattern recognition

4 Experiments

5 Conclusion
Unsupervised inference

Setting

- Given data about the genes proteins...
- Infer the edges between genes and proteins
- Note that the graph is considered completely unknown in the inference process

Strategies for inference

- Model-based: fit a “model” involving a graph to the data
- Similarity-based: connect “similar” nodes
Unsupervised inference

Setting

- Given data about the genes proteins...
- Infer the edges between genes and proteins
- Note that the graph is considered completely unknown in the inference process

Strategies for inference

- Model-based: fit a “model” involving a graph to the data
- Similarity-based: connect “similar” nodes
Model-based approaches

Strategy

1. Define a model to explain the data with a graph
2. Fit the model to the data to infer a graph

Examples

- Dynamical system to model gene expression time series (boolean network, PDE, state-space models...)
- Statistical models where the graph represents conditional independence relationships among random variables (Bayesian networks, LASSO, ...)
Model-based approaches

Strategy

1. Define a model to explain the data with a graph
2. Fit the model to the data to infer a graph

Examples

- **Dynamical system** to model gene expression time series (boolean network, PDE, state-space models...)
- **Statistical models** where the graph represents conditional independence relationships among random variables (Bayesian networks, LASSO, ...)
### Model-based approaches

**Pros**
- **Best approach** if the model is correct and enough data are available
- **Interpretability** of the model
- **Inclusion of prior knowledge**

**Cons**
- **Specific** to particular data and networks
- **Needs a correct model!**
- **Difficult integration** of heterogeneous data
- **Often needs a lot of data** and long computation time
Similarity-based approaches

**Rationale**
Genes functionally related are likely to be co-regulated, co-localized, present in the same organisms...

**Strategy**
- Define a distance between proteins from the genomic data
- Predict an edge if the distance is below a threshold
Similarity-based approaches

Rationale

Genes functionally related are likely to be co-regulated, co-localized, present in the same organisms...

Strategy

1. Define a distance between proteins from the genomic data
2. Predict an edge if the distance is below a threshold
Similarity-based approaches

Rationale
Genes functionally related are likely to be co-regulated, co-localized, present in the same organisms...

Strategy
1. Define a distance between proteins from the genomic data
2. Predict an edge if the distance is below a threshold
We assume that each type of data (expression, sequences...) defines a distance between genes.

Many such distances exist (cf kernel methods).

Data integration is easily obtained by summing the distance to obtain an “integrated” distance.
Evaluation on metabolic network reconstruction

- The known metabolic network of the yeast involves 769 proteins.
- Predict edges from distances between a variety of genomic data (expression, localization, phylogenetic profiles, interactions).
What went wrong?

Limitations

- Is the assumption that “similar proteins are connected” correct and sufficient?
- Is the Euclidean distance the “correct” way to compare genomic data?
- Perhaps the network inferred is interesting, but not related to the metabolic network?
Outline

1. Motivation
2. Unsupervised inference
3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition
4. Experiments
5. Conclusion
In actual applications, we know in advance parts of the network to be inferred, the problem is to add/remove nodes and edges using genomic data as side information.

- Given genomic data and the currently known network...
- Infer missing edges between current nodes and additional nodes.
Outline

1 Motivation

2 Unsupervised inference

3 Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4 Experiments

5 Conclusion
Metric learning

Idea

- The direct similarity-based method fails because the distance metric used might not be adapted to the inference of the targeted protein network.
- Solution: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
**Metric learning**

**Idea**

- The direct similarity-based method fails because the distance metric used might not be adapted to the inference of the targeted protein network.
- Solution: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
The direct similarity-based method fails because the distance metric used might not be adapted to the inference of the targeted protein network.

Solution: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
Idea

- The direct similarity-based method fails because the distance metric used might not be adapted to the inference of the targeted protein network.
- Solution: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
Embed both the graph and the genomic data in Hilbert spaces.

- Find subspaces in the Hilbert spaces where the graph distance and the genomic data distance match (kernel CCA).
- Use the metric of the genomic data subspace for network inference with the direct method.
Embed both the graph and the genomic data in Hilbert spaces.

Find subspaces in the Hilbert spaces where the graph distance and the genomic data distance match (kernel CCA).

Use the metric of the genomic data subspace for network inference with the direct method.
Embed both the graph and the genomic data in Hilbert spaces.

Find subspaces in the Hilbert spaces where the graph distance and the genomic data distance match (kernel CCA)

Use the metric of the genomic data subspace for network inference with the direct method.
Embed both the graph and the genomic data in Hilbert spaces.

Find subspaces in the Hilbert spaces where the graph distance and the genomic data distance match (kernel CCA).

Use the metric of the genomic data subspace for network inference with the direct method.

Metric learning by kernel CCA (Yamanishi et al., 2004)
Metric learning by kernel CCA (Yamanishi et al., 2004)

- **Embed** both the graph and the genomic data in Hilbert spaces.
- **Find** subspaces in the Hilbert spaces where the graph distance and the genomic data distance match (kernel CCA).
- Use the **metric of the genomic data subspace** for network inference with the direct method.
Kernel metric learning (V. and Yamanishi, 2005)

Kernel metric learning

- **Criterion**: connected points should be near each other after mapping to a new $d$-dimensional Euclidean space.
- Add **regularization** to deal with high dimensions.
- Mapping $f(x) = (f_1(x), \ldots, f_d(x))$ with:

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

- Interpolates between (kernel) PCA ($\lambda = \infty$) and graph embedding ($\lambda = 0$).
- Equivalent to a generalized eigenvalue problem.
Solves an important question of the similarity-based approach: which distance should be used?

Virtually any algorithm for distance metric learning can be used.

But... do we really need to follow the similarity-based approach to infer graphs?
Metric learning: Summary

- Solves an important question of the similarity-based approach: **which distance should be used?**
- Virtually any algorithm for **distance metric learning** can be used
- But... do we **really** need to follow the similarity-based approach to infer graphs?
1 Motivation

2 Unsupervised inference

3 Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4 Experiments

5 Conclusion
Matrix completion

Idea

- **Goal**: Fill **missing entries in the adjacency matrix directly**
- Use genomic data matrix (similarity/distance) as side information
Method

\[ \mathcal{M} \text{ is the set of matrices obtained when missing entries are filled} \]

\[ \mathcal{D} \text{ is the set of spectral variants of the genomic data matrix} \]

Find the completed matrix \( M \) by solving

\[ \min_{M \in \mathcal{M}, D \in \mathcal{D}} KL(D, M) \]
Matrix completion by kernel matrix regression (Yamanishi and V., 2007)

Method

- Embed the genomic data to a Hilbert space $\mathcal{H}$
- Formulate the problem as a bivariate regression problem:

$$M(x, y) = u(x)^\top u(y) + \epsilon,$$

where $u : \mathcal{H} \rightarrow \mathbb{R}^d$.

- A variant of the EM algorithm, using the Euclidean geometry instead of the information geometry.
Matrix completion: Summary

- **Algebraic** formulation of the problem
- Use specific **geometries** of the set of matrices (information geometry, Frobenius distances)
- However **not really motivated** by biological motivations
- In fact **closely related** to metric learning approaches (central role of **spectral** decomposition)
Outline

1. Motivation
2. Unsupervised inference
3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition
4. Experiments
5. Conclusion
Pattern recognition

- Input variables $\mathbf{x} \in \mathcal{X}$, Output $y \in \{-1, 1\}$.
- Training set $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.
- Goal: learn a function $f : \mathcal{X} \mapsto \{-1, 1\}$
- Many powerful algorithms! Logistic regression, nearest neighbors, ANN, decision trees, SVM
Input variables $\mathbf{x} \in \mathcal{X}$, Output $y \in \{-1, 1\}$.

Training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.

Goal: learn a function $f: \mathcal{X} \mapsto \{-1, 1\}$

Many powerful algorithms! Logistic regression, nearest neighbors, ANN, decision trees, SVM
Formulation and basic issue

- A pair can be connected (1) or not connected (-1).
- From the known subgraph we can extract examples of connected and non-connected pairs.
- However, the genomic data characterize individual proteins; we need to work with pairs of proteins instead!
A pair can be **connected** (1) or **not connected** (-1).

From the known subgraph we can **extract examples** of connected and non-connected pairs.

However the genomic data characterize **individual** proteins; we need to work with **pairs** of proteins instead!
Formulation and basic issue

- A pair can be connected (1) or not connected (-1)
- From the known subgraph we can extract examples of connected and non-connected pairs
- However the genomic data characterize individual proteins; we need to work with pairs of proteins instead!
**Tensor product SVM (Ben-Hur and Noble, 2006)**

- **Intuition**: a pair \((A, B)\) is similar to a pair \((C, D)\) if:
  - \(A\) is similar to \(C\) and \(B\) is similar to \(D\), or...
  - \(A\) is similar to \(D\) and \(B\) is similar to \(C\)

- Formally, define a similarity between pairs from a similarity between individuals by

\[
K_{TPPK} ((a, b), (c, d)) = K(a, c)K(b, d) + K(a, d)K(b, c) .
\]

- If \(K\) is a positive definite kernel for individuals then \(K_{TPPK}\) is a p.d. kernel for pairs which can be used by SVM

- This amounts to representing a pair \((a, b)\) by the symmetrized tensor product:

\[
(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .
\]
**Intuition:** a pair \((A, B)\) is similar to a pair \((C, D)\) if:
- \(A\) is similar to \(C\) and \(B\) is similar to \(D\), or...
- \(A\) is similar to \(D\) and \(B\) is similar to \(C\)

**Formally,** define a similarity between pairs from a similarity between individuals by

\[
K_{TPPK} \left( ((a, b), (c, d)) \right) = K(a, c)K(b, d) + K(a, d)K(b, c) .
\]

- If \(K\) is a positive definite kernel for individuals then \(K_{TPPK}\) is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair \((a, b)\) by the symmetrized tensor product:

\[
(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .
\]
Tensor product SVM (Ben-Hur and Noble, 2006)

- **Intuition**: a pair \((A, B)\) is similar to a pair \((C, D)\) if:
  - \(A\) is similar to \(C\) and \(B\) is similar to \(D\), or...
  - \(A\) is similar to \(D\) and \(B\) is similar to \(C\)

- **Formally**, define a similarity between pairs from a similarity between individuals by

\[
K_{TPPK} \left( ((a, b), (c, d)) \right) = K(a, c)K(b, d) + K(a, d)K(b, c) .
\]

- If \(K\) is a positive definite kernel for individuals then \(K_{TPPK}\) is a p.d. kernel for pairs which can be used by SVM

- This amounts to representing a pair \((a, b)\) by the symmetrized tensor product:

\[
(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .
\]
**Intuition:** a pair \((A, B)\) is similar to a pair \((C, D)\) if:

- \(A - B\) is similar to \(C - D\), or...
- \(A - B\) is similar to \(D - C\).

**Formally,** define a similarity between pairs from a similarity between individuals by

\[
K_{MLPK} \left( ((a, b), (c, d)) \right) = (K(a, c) + K(b, d) - K(a, c) - K(b, d))^2 .
\]

If \(K\) is a positive definite kernel for individuals then \(K_{MLPK}\) is a p.d. kernel for pairs which can be used by SVM.

This amounts to representing a pair \((a, b)\) by the **symmetrized difference**:

\[
(a, b) \rightarrow (a - b)^2 .
\]
**Intuition**: a pair \((A, B)\) is similar to a pair \((C, D)\) if:
- \(A - B\) is similar to \(C - D\), or...
- \(A - B\) is similar to \(D - C\).

**Formally**, define a similarity between pairs from a similarity between individuals by

\[
K_{MLPK} \left( ((a, b), (c, d)) \right) = (K(a, c) + K(b, d) - K(a, c) - K(b, d))^2.
\]

If \(K\) is a positive definite kernel for individuals then \(K_{MLPK}\) is a p.d. kernel for pairs which can be used by SVM.

This amounts to representing a pair \((a, b)\) by the symmetrized difference:

\[
(a, b) \rightarrow (a - b)^\otimes 2.
\]
**Intuition:** a pair \((A, B)\) is similar to a pair \((C, D)\) if:
- \(A - B\) is similar to \(C - D\), or...
- \(A - B\) is similar to \(D - C\).

**Formally,** define a similarity between pairs from a similarity between individuals by

\[
K_{MLPK} \left( ((a, b), (c, d)) \right) = (K(a, c) + K(b, d) - K(a, c) - K(b, d))^2.
\]

If \(K\) is a positive definite kernel for individuals then \(K_{MLPK}\) is a p.d. kernel for pairs which can be used by SVM

This amounts to representing a pair \((a, b)\) by the symmetrized difference:

\[
(a, b) \rightarrow (a - b)^\otimes 2.
\]
Remarks about pattern recognition for pairs

Pros

- The **objective function** is exactly what we want (discriminate between connected and non-connected pairs)
- We can use **state-of-the-art powerful algorithms** for graph inference (e.g., SVM)

Cons

- We need to deduce an **embedding for pairs** from data about individuals.
- There are many training examples \(N(N - 1)/2\) which can be a problem of pattern recognition algorithms in terms of computation time and memory.
- The result is a **global** model over the graph; however the presence or absence of a connection may also depend on the “position” of the connection in the graph.
## Remarks about pattern recognition for pairs

### Pros
- **The objective function** is exactly what we want (discriminate between connected and non-connected pairs)
- **We can use state-of-the-art powerful algorithms** for graph inference (e.g., SVM)

### Cons
- We need to deduce an **embedding for pairs** from data about individuals.
- There are **many** training examples \((N(N-1)/2)\) which can be a problem of pattern recognition algorithms in terms of computation time and memory
- The result is a **global** model over the graph; however the presence or absence of a connection may also depend on the “position” of the connection in the graph.
Outline

1. Motivation
2. Unsupervised inference
3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition
4. Experiments
5. Conclusion
Motivation: define specific models for each target node to discriminate between its neighbors and the others.

Treat each node independently from the other. Then combine predictions for ranking candidate edges.
Motivation: define specific models for each target node to discriminate between its neighbors and the others. Treat each node independently from the other. Then combine predictions for ranking candidate edges.
The LOCAL model
The LOCAL model: training edges
The LOCAL model: testing edges
The LOCAL model: learning
The LOCAL model: learning
The LOCAL model: learning
The LOCAL model: learning
The LOCAL model: learning
The LOCAL model: decision boundary
The LOCAL model: testing
The LOCAL model: testing
The LOCAL model: Predictions
The LOCAL model: target graph
The LOCAL model: Two correct edges, one error
The LOCAL model: Do same for each learning node
The LOCAL model: Do same for each learning node
The LOCAL model: Do same for each learning node
The LOCAL model: Do same for each learning node
The LOCAL model: Do same for each learning node
The LOCAL model: Do same for each learning node
Local predictions: pros and cons

**Pros**
- Allow **very different models** for nearby nodes on the graph
- **Faster** to train $n$ models with $n$ examples than 1 model with $n^2$ examples
- No need for tricky embedding of pairs: each model works at the level of individuals.

**Cons**
- Few positive examples available for some nodes
- We must rank pairs based on scores obtained on different models $\implies$ scores must be calibrated.
- If we have **two new proteins**, no simple way to predict an edge between them.
Local predictions: pros and cons

**Pros**
- Allow very different models for nearby nodes on the graph
- Faster to train $n$ models with $n$ examples than 1 model with $n^2$ examples
- No need for tricky embedding of pairs: each model works at the level of individuals.

**Cons**
- Few positive examples available for some nodes
- We must rank pairs based on scores obtained on different models $\Rightarrow$ scores must be calibrated.
- If we have two new proteins, no simple way to predict an edge between them.
Outline

1. Motivation

2. Unsupervised inference

3. Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4. Experiments

5. Conclusion
Experiments

Network
- Metabolic network (668 vertices, 2782 edges)
- Protein-protein interaction network (984 vertices, 2438 edges)

Data (yeast)
- Gene expression (157 experiments)
- Phylogenetic profile (145 organisms)
- Cellular localization (23 intracellular locations)
- Yeast two-hybrid data (2438 interactions among 984 proteins)

Method
- 5-fold cross-validation
- Predict edges between test set and training set
Results: protein-protein interaction

![Graph showing protein-protein interaction results](image-url)

- Ratio of true positives
- Ratio of false positives
- False discovery rate

Lines represent different methods:
- Direct
- kML
- kCCA
- em
- local
- Pkernel

J.-P. Vert (ParisTech)
Results: metabolic gene network

![Graph showing results for metabolic gene network with comparison of different methods: Direct, kML, kCCA, em, local, Pkernel. The graphs display the ratio of false positives against the ratio of true positives, and the false discovery rate against the ratio of true positives. Each method is represented by a different line color.](image-url)
Local SVM, protein-protein interaction network.
Results: effect of data integration

Local SVM, metabolic gene network.
Experiments: Summary

- **Supervised approaches** work much better than the baseline direct approach.
- **Data integration** is easy and very powerful.
- Good results obtained on two apparently **very different networks** (metabolic, physical interactions).
- The **LOCAL method** wins the benchmark competition.
Applications: missing enzyme prediction

Prediction of missing enzyme genes in a bacterial metabolic network

Reconstruction of the lysine-degradation pathway of *Pseudomonas aeruginosa*

Yoshihiro Yamanishi¹, Hisaaki Mihara², Motoharu Osaki², Hisashi Muramatsu³, Nobuyoshi Esaki², Tetsuya Sato¹, Yoshiyuki Hizukuri¹, Susumu Goto¹ and Minoru Kanehisa¹

1 Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan
2 Division of Environmental Chemistry, Institute for Chemical Research, Kyoto University, Japan
3 Department of Biology, Graduate School of Science, Osaka University, Japan

![Gene Location Diagram]

![Predicted Gene Network Diagram]
Applications: missing enzyme prediction
Prediction of nitrogen metabolism-related genes in *Anabaena* by kernel-based network analysis

Shinobu Okamoto\(^1\)*, Yoshihiro Yamanishi\(^1\), Shigeki Ehira\(^2\), Shuichi Kawashima\(^3\), Koichiro Tonomura\(^1\)** and Minoru Kanehisa\(^1\)

\(^1\) Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan
\(^2\) Department of Biochemistry and Molecular Biology, Faculty of Science, Saitama University, Saitama, Japan
\(^3\) Human Genome Center, Institute of Medical Science, University of Tokyo, Meguro, Japan
Determinition of the role of the bacterial peptidase PepF by statistical inference and further experimental validation

Liliana LOPEZ KLEINE\textsuperscript{1,2}, Alain TRUBUI\textsuperscript{1}, Véronique MONNET\textsuperscript{2}

\textsuperscript{1}Unité de Mathématiques et Informatiques Appliquées. INRA Jouy en Josas 78352, France.
\textsuperscript{2}Unité de Biochimie Bactérienne. INRA Jouy en Josas 78352, France.
1 Motivation

2 Unsupervised inference

3 Supervised inference
   - Metric learning
   - Matrix completion
   - Global pattern recognition
   - Local pattern recognition

4 Experiments

5 Conclusion
Take-home messages

- When the network is known in part, supervised methods can be more adapted than unsupervised ones.
- A variety of methods have been investigated recently (metric learning, matrix completion, pattern recognition); the current winner on our benchmarks (metabolic network and PPI network) is the local pattern recognition approach.
- It reaches high performance on the benchmarks: 45% of all true edges of the metabolic gene network are retrieved at a FDR below 50% (for the yeast).
- These methods:
  - work for any network
  - work with any data
  - can integrate heterogeneous data, which strongly improves performance
People I need to thank

- Yoshihiro Yamanishi, Minoru Kanehisa (Univ. Kyoto): kCCA, kML
- Jian Qian, Bill Noble (Univ. Washington): pairwise SVM
- Kevin Bleakley, Gerard Biau (Univ. Montpellier): local SVM