Semi-supervised learning from Complex Data

Michelangelo Ceci
University of Bari, Bari, Italy
Jožef Stefan Institute, Ljubljana, Slovenia

Supervised VS Semi-supervised Learning in predictive tasks

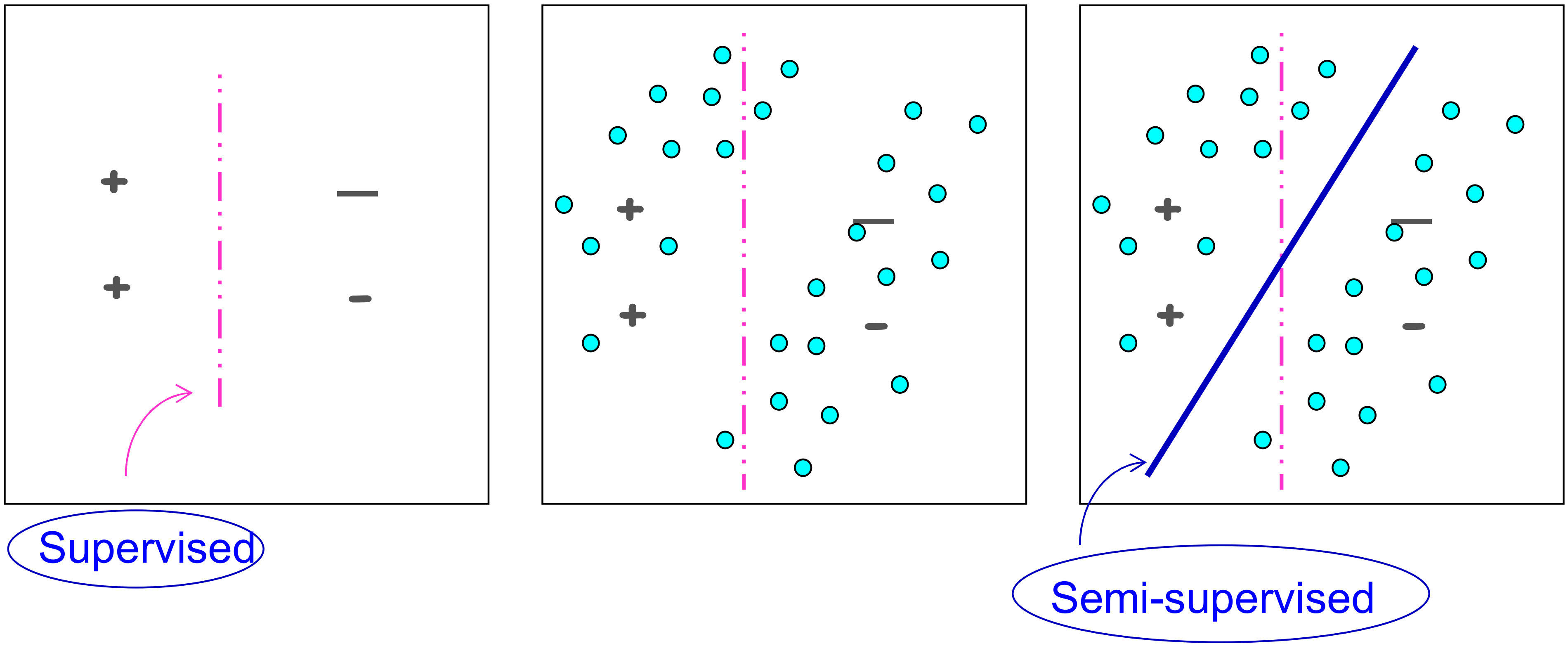
Supervised learning:

- Only labeled data are used to build the predictive model. Discards large amount of information potentially conveyed by unlabeled instances.

Semi-supervised learning:

- Both labeled & unlabeled data are used to build the predictive model.

Why semi-supervised learning?



Supervised

Semi-supervised

Why semi-supervised learning?

- Philosophical motivation:

  Human brain can exploit unlabeled data.

- Pragmatic motivation:

  Unlabeled data is usually cheaper to collect w.r.t. labeled data.

Why semi-supervised learning?

Labeled training data is scarce and expensive

- E.g., experiments in computational biology
- Need for expert knowledge
- Tedious and time consuming

Unlabeled instances are abundant and cheap

- Extract vectorized maps from satellite images
- Assess primary structure of proteins from DNA/RNA

Semi-supervised learning:
Inductive vs Transductive settings

Supervised learning:

- Only labeled data are used to build the predictive model. Discards large amount of information potentially conveyed by unlabeled instances.

Semi-supervised learning:

- Both labeled & unlabeled data are used to build the predictive model.
  - Transductive setting: the learned model can be applied to make predictions **only on the unlabeled instances known/observed** during the training phase.
  - Inductive setting: the learned model can be applied to make predictions on **any unlabeled instance**, either known/observed or unknown/unseen during the training phase.

Semi-supervised learning: Inductive vs Transductive settings

The difference is also clear in the experimental protocol:

- L: number of labelled cases
- U: number of unlabelled cases
- N: number of examples (possibly not available during learning)

- Transductive setting N=U+L: the training set comprises of N examples, L of which are labeled. Performance evaluated in predicting U = N – L unlabeled examples.

- Inductive setting N >> U+L: the training set comprises of L+U examples. Performance evaluated in predicting N-L-U unlabeled examples (or, in some cases, N-L examples).

Semi-supervised / Transductive learning: early references

- Transductive learning was used for the first time by Vapnik (Vapnik & Chervonenkis, 1974; Vapnik & Sterin, 1977)

- An early instance of transduction (albeit without explicitly considering it as a concept) was already proposed by Hartley and Rao (1968), who suggested a combinatorial optimization on the labels of the test points in order to maximize the likelihood of their model

- Interest for transductive learning increased in the 1990s, mostly due to applications in text classification

Semi-supervised / Transductive learning: early references

Semi-supervised learning (Chapelle, Schölkopf, Zien 2006)

https://www.molgen.mpg.de/3659531/MITPress--SemiSupervised-Learning.pdf

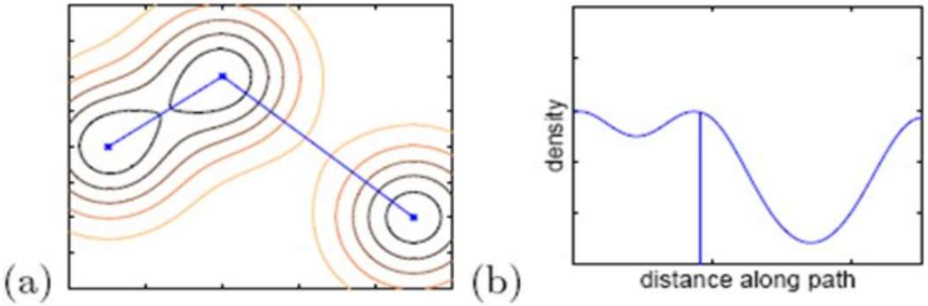
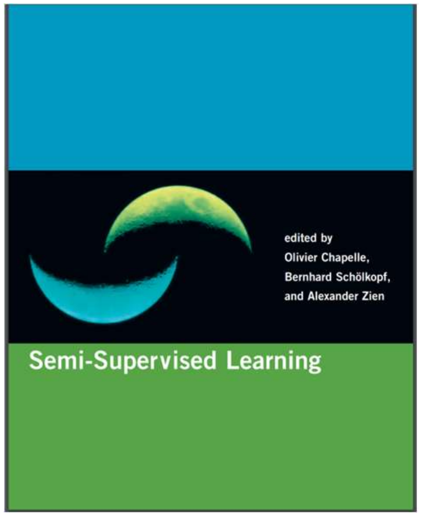


Figure 1: Optimal connecting curves are well approximated by paths of short distance edges on a graph.



Smoothness assumption in Supervised Learning

If two points x_1 and x_2 are close, then so should be the corresponding outputs y_1, y_2

Without such assumption, it would never be possible to generalize from a finite training set to a set of possibly infinite unseen test cases.

The application of this assumption is evident in similarity-based learning:

- Training instances are stored in memory and a similarity metric is used to compare new instances to those stored/known.
- New instances are classified according to the closest examples in memory.

Smoothness assumption in Semi-Supervised Learning

If two points x_1 and x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2

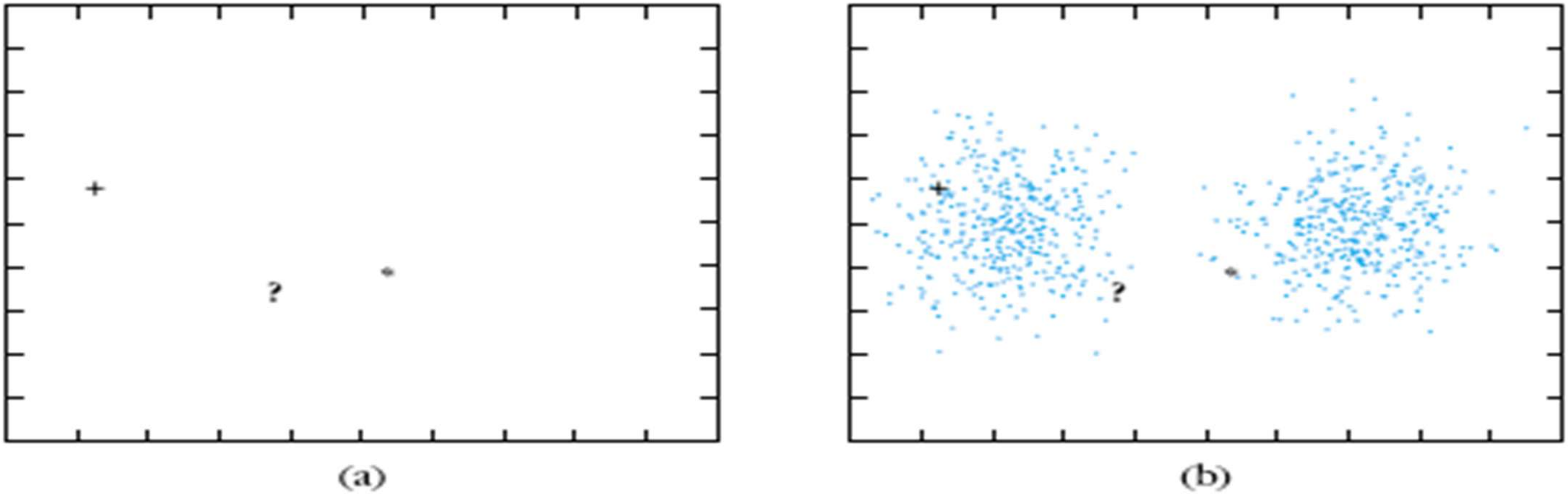
- The label function is smoother in high-density regions than in low-density regions.

- This assumption entails that if two points are separated by a low-density region, then their outputs need not to be close.

- It is also called label smoothness assumption.

Smoothness assumption in Semi-Supervised Learning

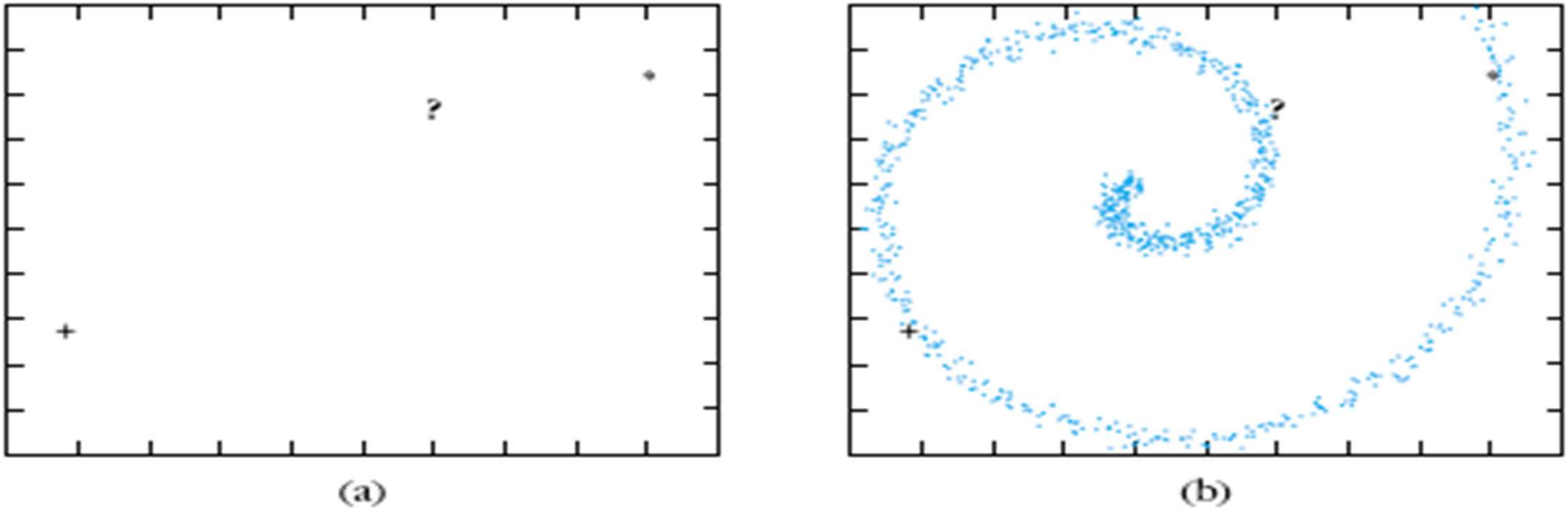
Closeness between points is not a decisive factor, if considered by itself. It has to be considered in the context of the underlying distribution.



(a) The unknown point, denoted by “?”, is classified in the same class as point “*”. (b) The setup after a number of unlabeled data have been provided, which leads us to reconsider our previous classification decision.

Smoothness assumption in Semi-Supervised Learning

Closeness between points is not a decisive factor, if considered by itself. It has to be considered in the context of the underlying distribution.



(a) The unknown point, denoted by "?", is classified in the same class as point "*". (b) The setup after a number of unlabeled data have been provided, which leads us to reconsider our previous classification decision.

Cluster assumption

- If points are in the same cluster, they are likely to be of the same class.

- Idea: run a clustering algorithm and use the labeled points to assign a class to each cluster. This is in fact one of the earliest forms of semi-supervised learning.

- The cluster assumption can be seen as a special case of the semi-supervised smoothness assumption, when clusters are defined by considering only high-density regions.

- Low density separation: the decision boundary should lie in a low-density region.

Semi-Supervised Learning: Basic Algorithms

- Self Training
- Generative Models
- S3VMs
- Graph-Based Algorithms
- Deep Learning

Self-training algorithm

- Self-training algorithm:
- Train f from the set of labeled examples L
- Predict on x ∈ U (unlabeled data)
- Add a few most confident (x, f(x)) to L
- Repeat

Pros and cons of self-training

PROS

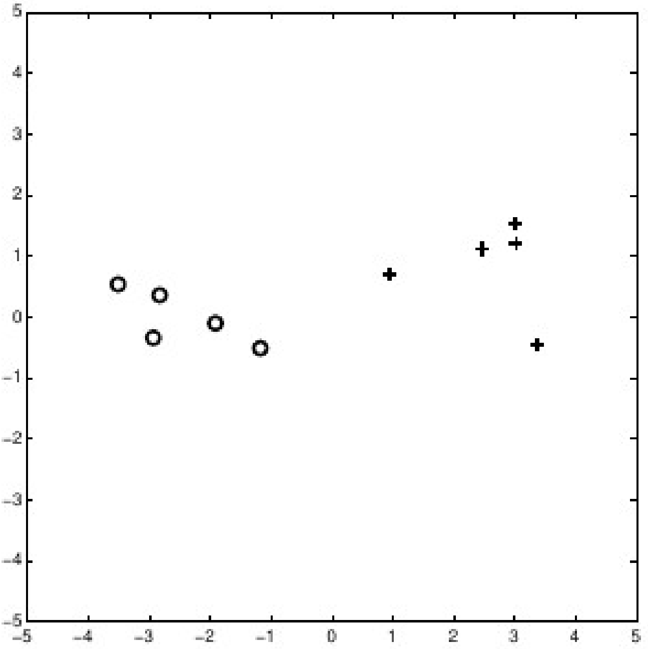
- The simplest semi-supervised learning method.
- A wrapper method, applies to existing (complex) classifiers.
- Often used in real tasks like natural language processing.

CONS

- Early mistakes could reinforce themselves
- Cannot say too much in terms of convergence.
  - But there are special cases when self-training is equivalent to the Expectation-Maximization (EM) algorithm.

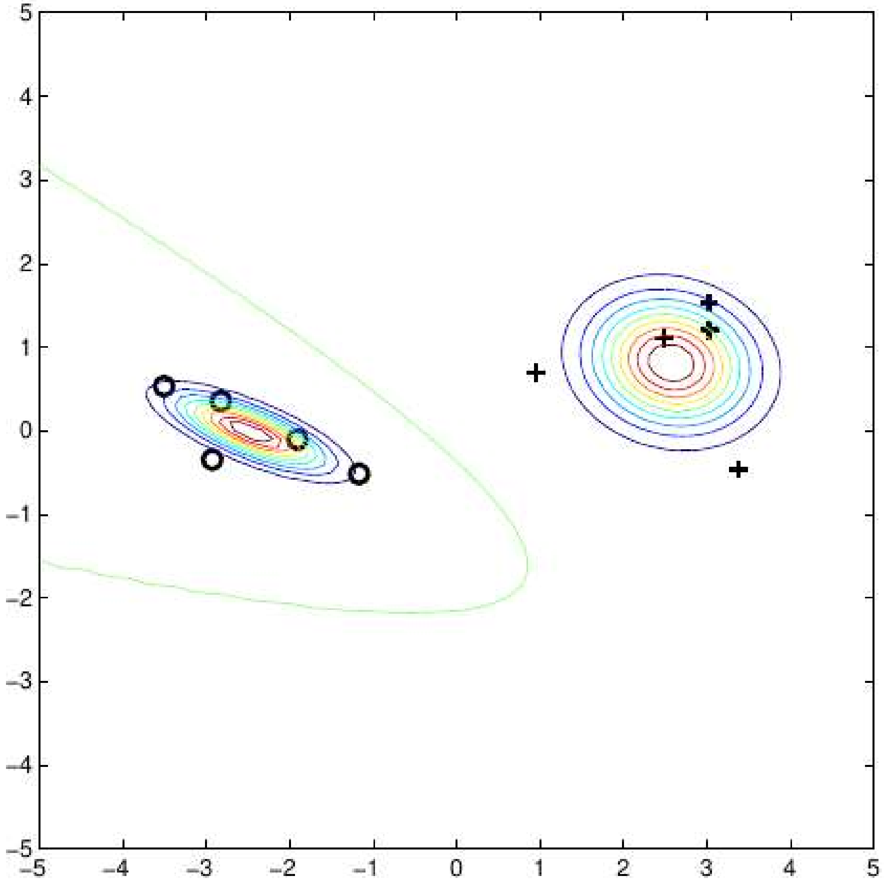
Generative Models

Labeled data:



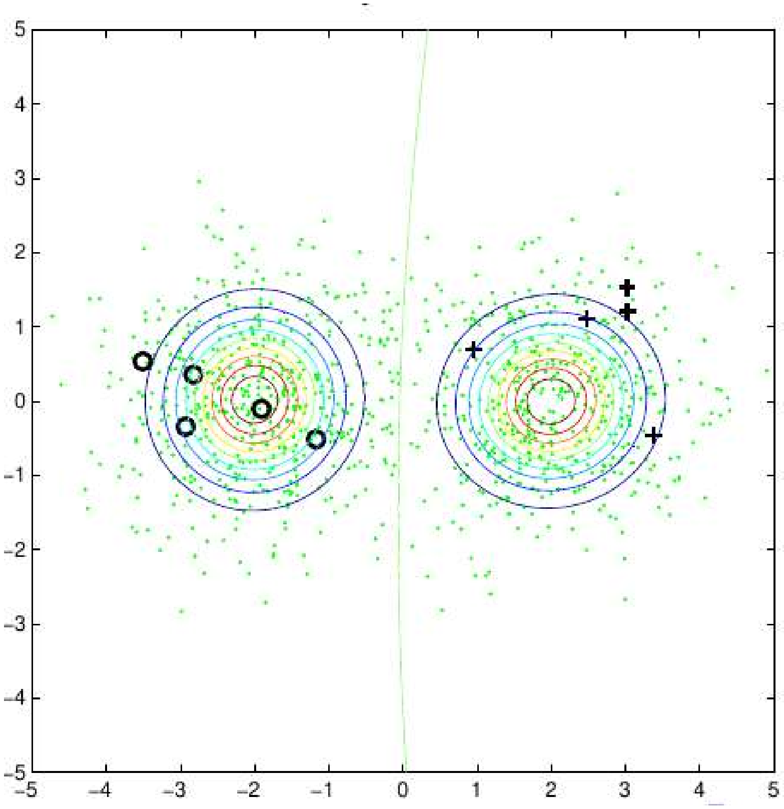
Assuming each class has a Gaussian distribution, what is the decision boundary?

Generative Models



Generative Models

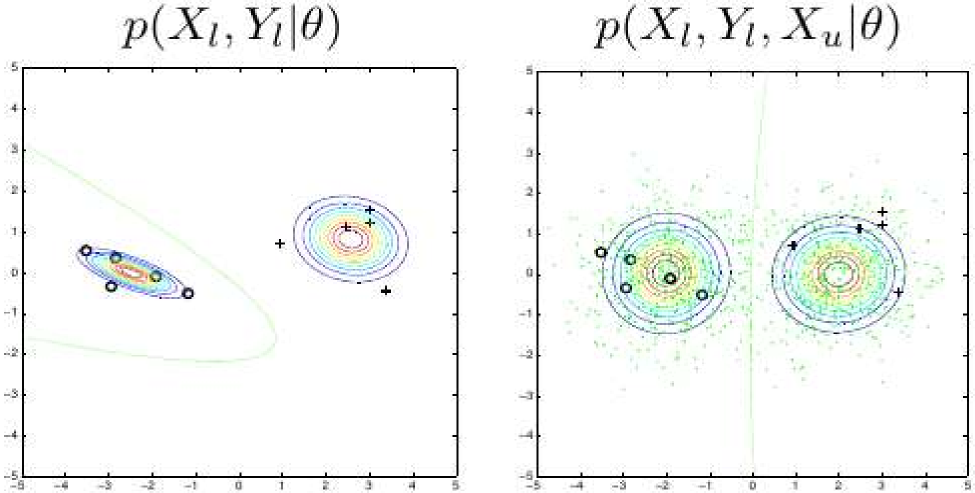
Adding unlabeled data:



With unlabeled data, the most likely model and its decision boundary change

Generative Models

They are different because they maximize different quantities



$p(X_l, Y_l|\theta)$

$p(X_l, Y_l, X_u|\theta)$

$$p(X_l, Y_l, X_u|\theta) = \sum_{Y_u} p(X_l, Y_l, X_u, Y_u|\theta)$$

Find the maximum likelihood estimate (MLE) of θ, the maximum a posteriori (MAP) estimate, or the Bayesian

Generative Models: pros and cons

Pros:

- Clear, well-studied probabilistic framework
- Can be extremely effective, if the model is close to correct

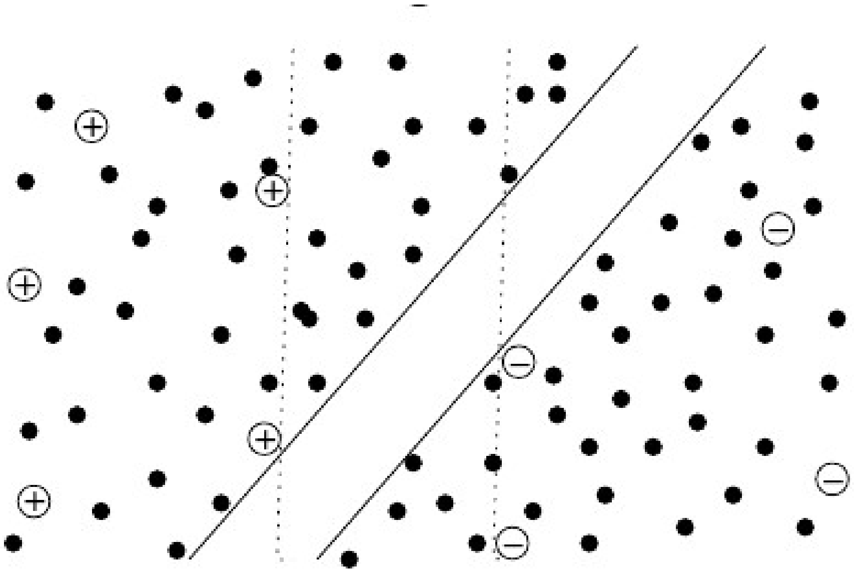
Cons:

- Often difficult to verify the correctness of the model
- Model identifiability
- EM local optima
- Unlabeled data may hurt if generative model is wrong

Semi-supervised Support Vector Machines

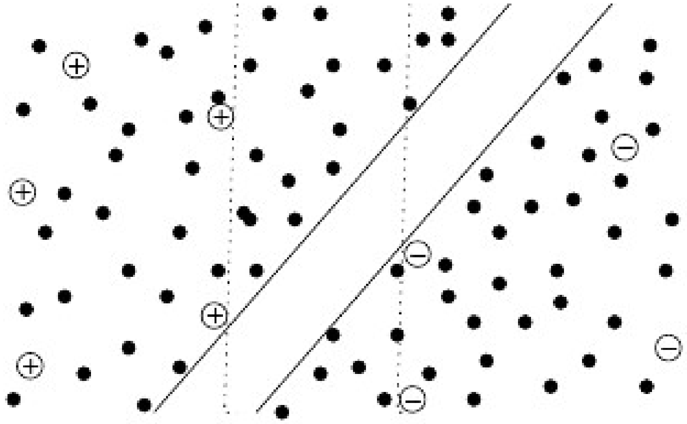
Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)

Maximizes "unlabeled data margin"



K.P. Bennett and A. Demiriz. Semi-supervised support vector machines. In Advances in Neural Information processing systems, pages 368–374, 1999

Semi-supervised Support Vector Machines



S3VM idea:

- Enumerate all $2^{|U|}$ possible labeling of U
- Build one standard SVM for each labeling (and x)
- Pick the SVM with the largest margin

$$\min_{f} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \lambda_1 \|h\|^2_{\mathcal{H}_K} + \lambda_2 \sum_{i=l+1}^{n} (1 - |f(x_i)|)_+$$

the third term prefers unlabeled points outside the margin. Equivalently, the decision boundary f = 0 wants to be placed so that there is few unlabeled data near it.

Semi-supervised Support Vector Machines: pros and cons

Pros:

- Applicable wherever SVMs are applicable
- Clear mathematical framework

Cons:

- Optimization difficult
- Can be trapped in bad local optima
- More modest assumption than generative model or graph-based methods, potentially lesser gain

Graph-based semi-supervised learning: pros and cons

Assumption

A graph is given on the labeled and unlabeled data. Instances connected by heavy edge tend to have the same label.

Graph-based semi-supervised learning: pros and cons

The graph mincut problem:

Fix Y_l, find $Y_u \in \{0,1\}^{n-l}$ to minimize $\sum_{ij} w_{ij}|y_i - y_j|$

Or, equivalently:

$$\min_{Y \in \{0,1\}^n} \ \infty \sum_{i=1}^{l} (y_i - Y_{li})^2 + \sum_{ij} w_{ij}(y_i - y_j)^2$$

Combinatorial problem, but has polynomial time solution.

Graph-based semi-supervised learning: pros and cons

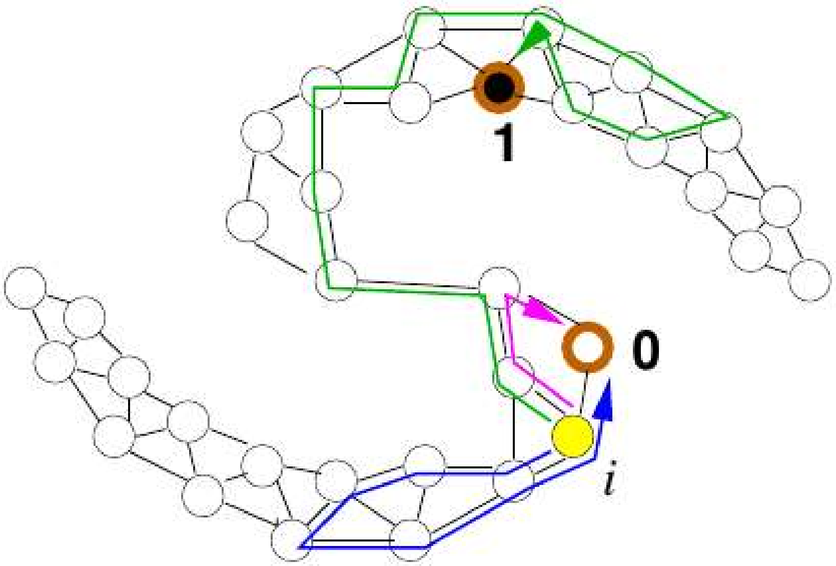
Random walk interpretation:

Randomly walk from node i to j with probability $\frac{w_{ij}}{\sum_k w_{ik}}$

Stop if we hit a labeled node

Compute the harmonic function

$$f = Pr(\text{hit label 1}|\text{start from } i)$$



Graph-based semi-supervised learning: pros and cons

Pros:

- Clear mathematical framework
- Performance is strong if the graph happens to fit the task
- Can be extended to directed graphs

Cons:

- Performance is bad if the graph is bad
- Sensitive to graph structure and edge weights

CNNs for Semi-Supervised Learning

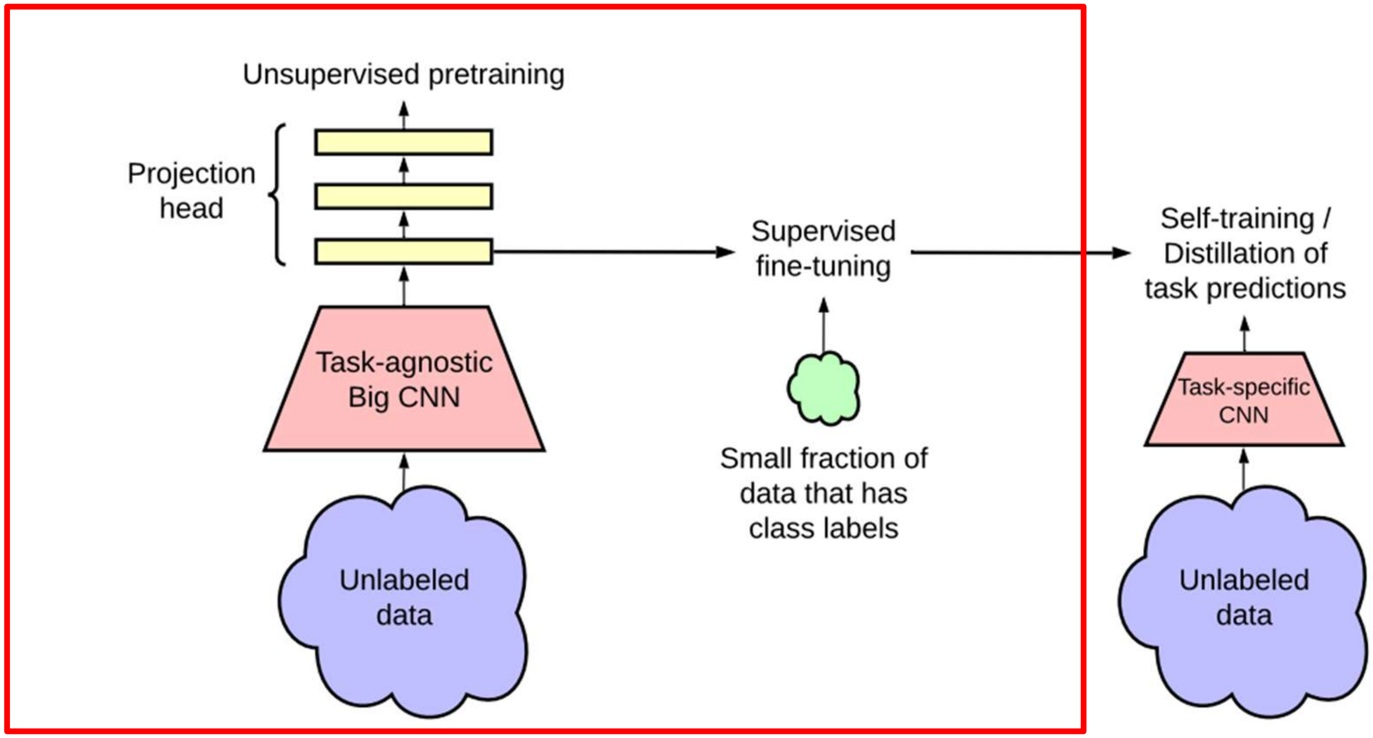


Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

Chen T., Kornblith S., Swersky K., Norouzi M., Hinton G. *Big self-supervised models are strong semi-supervised learners* (2020) Advances in Neural Information Processing Systems, 2020

Semi-Supervised Learning with Unsupervised Data Augmentation

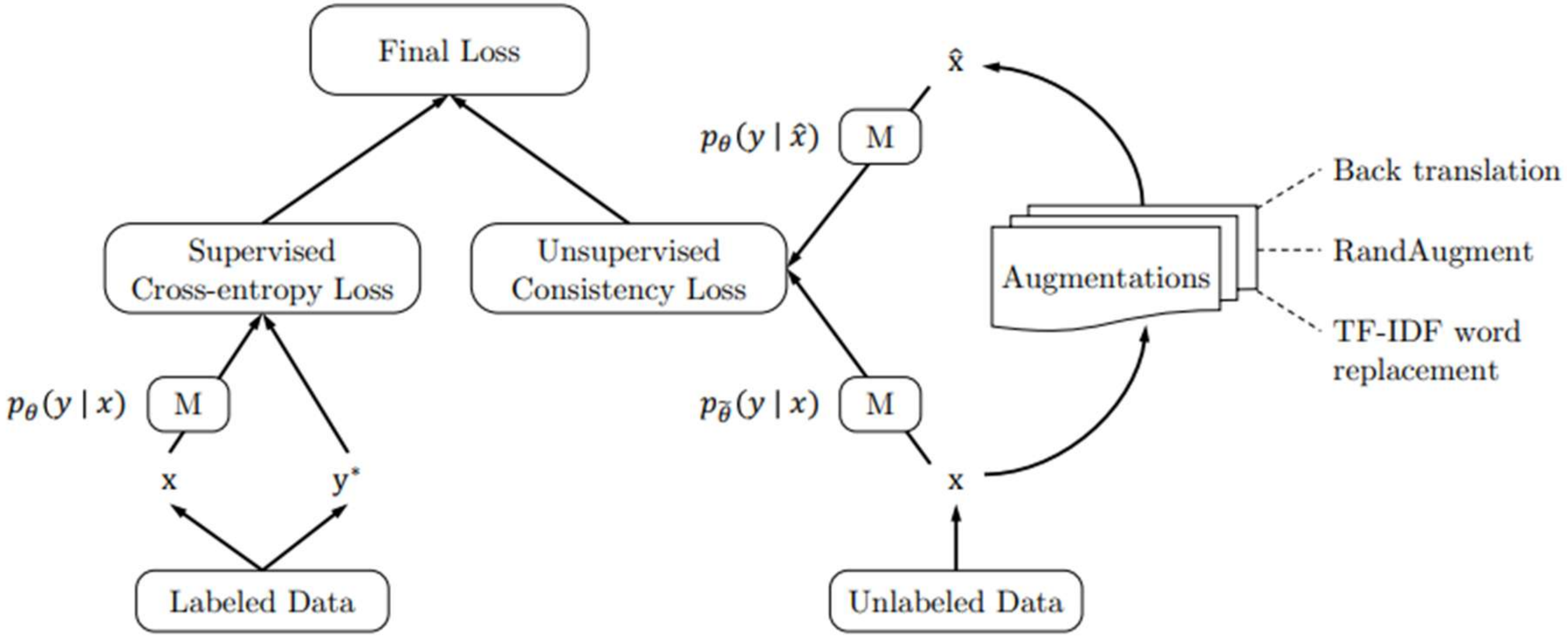


Figure 1: Training objective for UDA, where M is a model that predicts a distribution of y given x.

Xie Q., Dai Z., Hovy E., Luong M.-T., Le Q.V. Unsupervised data augmentation for consistency training (2020) *Advances in Neural Information Processing Systems*, 2020

Deep Semi-Supervised Learning

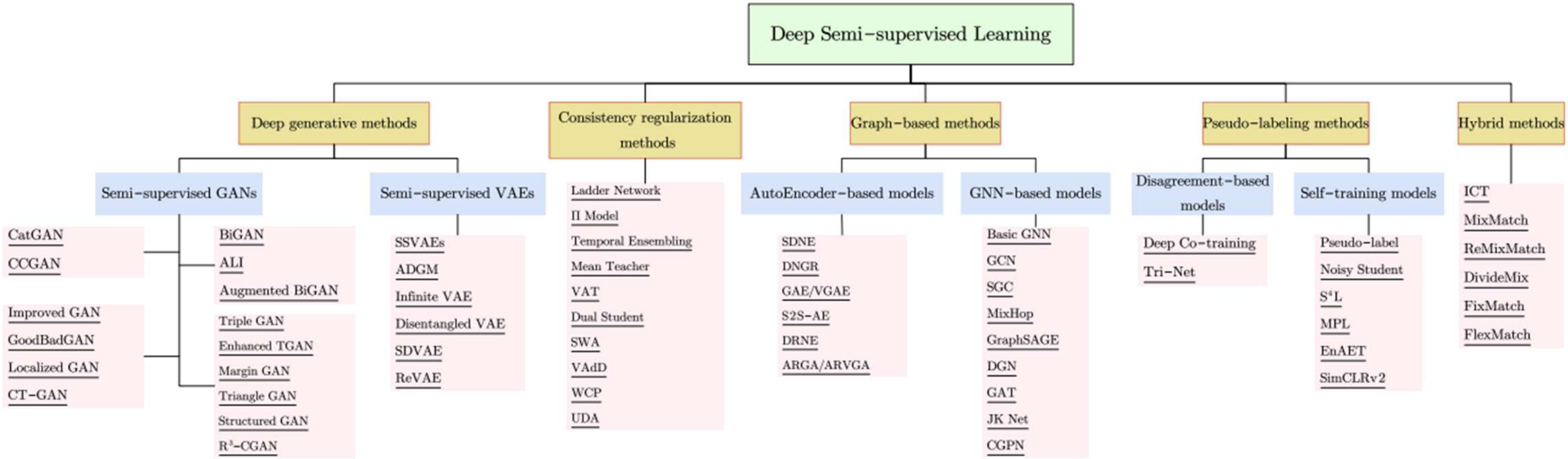


Fig. 1. The taxonomy of major deep semi-supervised learning methods based on loss function and model design.

$$\min_{\theta} \underbrace{\sum_{(x,y)\in X_L} \mathcal{L}_s(x,y,\theta)}_{\text{supervised loss}} + \alpha \underbrace{\sum_{x\in X_U} \mathcal{L}_u(x,\theta)}_{\text{unsupervised loss}} + \beta \underbrace{\sum_{x\in X} \mathcal{R}(x,\theta)}_{\text{regularization}}$$

Yang X., Song Z., King I., Xu Z. A Survey on Deep Semi-Supervised Learning(2023) *IEEE Transactions on Knowledge and Data Engineering*, 35 (9), pp. 8934 - 8954

Sources of complexity in real-world scientific domains

Output space: Structured output prediction, predicting more **complex outputs** than in classification/regression:

- Multi-target regression (MTR)
- Multi-label classification (MLC)
- Hierarchical multi-label classification (HMLC)

Input space: data not independently and identically distributed

- Classification/Regression/Link prediction in
  - Homogeneous Network data
  - Heterogeneous Network data
  - Relational data

Semi-supervised learning in Structured output prediction

Multi-target prediction

- Classification

	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	Yes	Blue	Rain
Example 2	2	FALSE	0.08	0.07	Yes	Green	Sun
Example 3	1	FALSE	0.08	0.07	Yes	Blue	Cloudy
Example 4	2	TRUE	0.49	0.69	Yes	Green	Sun
Example 5	3	TRUE	0.49	0.69	No	Blue	Sun
Example 6	4	FALSE	0.08	0.07	Yes	Red	Cloudy
…	…				…	…	…

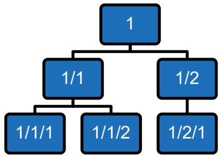
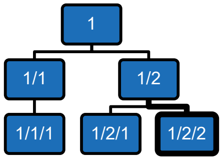
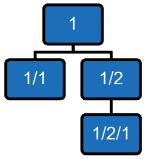
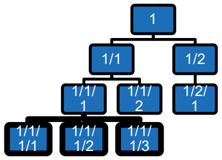
- Regression

	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	0.68	0.60	3.91
Example 2	2	FALSE	0.08	0.07	0.56	0.99	7.59
Example 3	1	FALSE	0.08	0.07	0.10	1.69	7.57
Example 4	2	TRUE	0.49	0.69	0.08	0.77	8.86
Example 5	3	TRUE	0.49	0.69	0.11	3.51	2.50
Example 6	4	FALSE	0.08	0.07	0.43	2.10	8.09
…	…				…	…	…

Multi-label classification

	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	A, B, D
Example 2	2	FALSE	0.08	0.07	B, D
Example 3	1	FALSE	0.08	0.07	A, D, E
Example 4	2	TRUE	0.49	0.69	D
…	…				…

Hierarchical multi-label classification

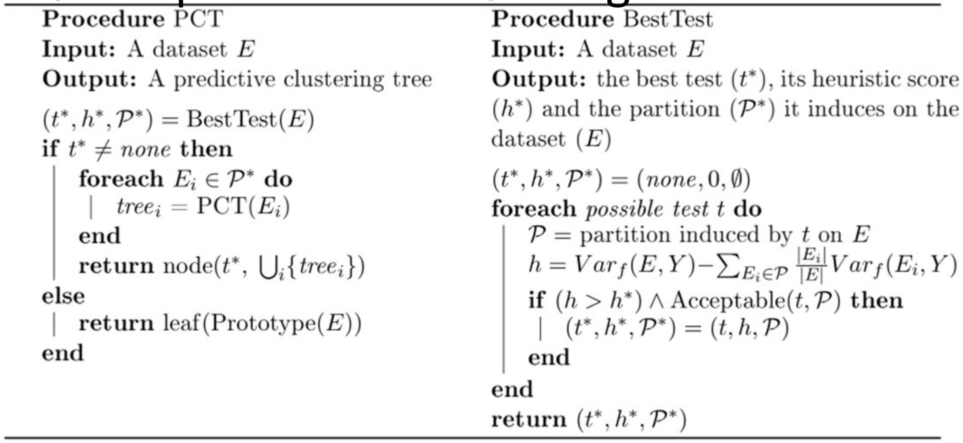
	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	
Example 2	2	FALSE	0.08	0.07	
Example 3	1	FALSE	0.08	0.07	
Example 4	2	TRUE	0.49	0.69	 1 1/1 1/2 1/1/1 1/1/2 1/2/1 1/1/1/1 1/1/1/2 1/1/1/3
…	…				…

e. g. Gene function prediction

Structured Output Prediction with Predictive Clustering Trees

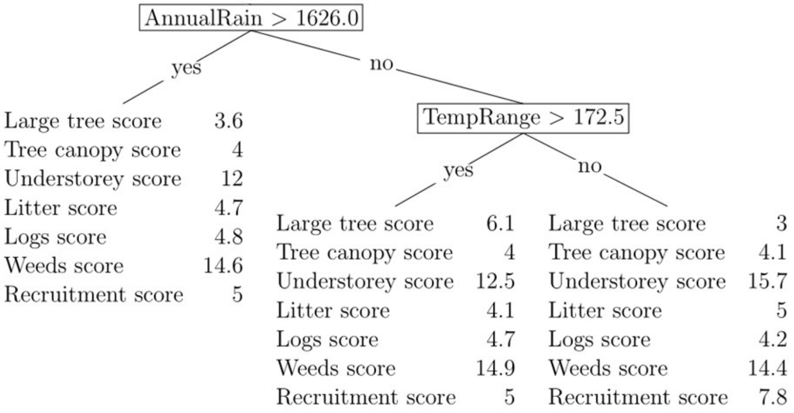
- Generalization of decision trees towards predicting structured outputs

The top-down induction algorithm for PCTs



```
Procedure PCT
Input: A dataset $E$
Output: A predictive clustering tree
$(t^*, h^*, \mathcal{P}^*) = \text{BestTest}(E)$
if $t^* \neq none$ then
    foreach $E_i \in \mathcal{P}^*$ do
        $tree_i = \text{PCT}(E_i)$
    end
    return node($t^*$, $\bigcup_i \{tree_i\}$)
else
    return leaf(Prototype($E$))
end
```

```
Procedure BestTest
Input: A dataset $E$
Output: the best test ($t^*$), its heuristic score ($h^*$) and the partition ($\mathcal{P}^*$) it induces on the dataset ($E$)
$(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
foreach possible test $t$ do
    $\mathcal{P}$ = partition induced by $t$ on $E$
    $h = Var_f(E, Y) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var_f(E_i, Y)$
    if $(h > h^*) \wedge \text{Acceptable}(t, \mathcal{P})$ then
        $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
    end
end
return $(t^*, h^*, \mathcal{P}^*)$
```



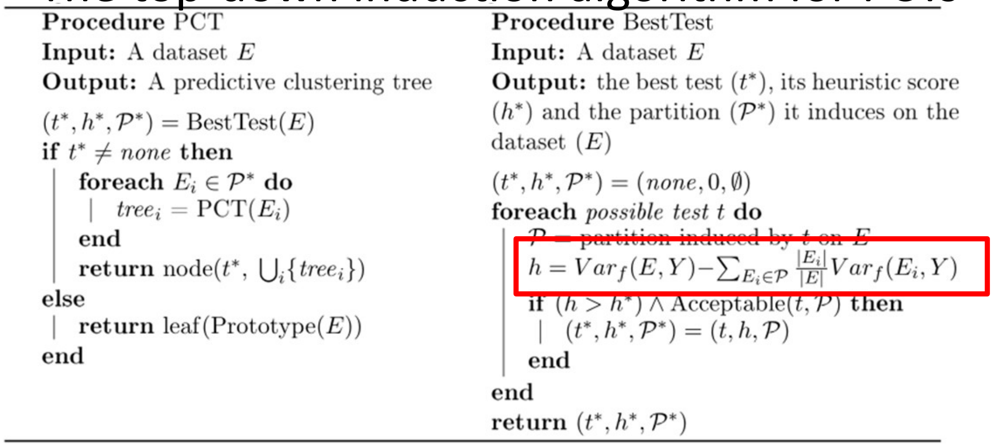
X is descriptive space, Y is target space, and E is a set of labeled examples

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first-order logical decision trees. *Artificial intelligence*.

Predictive Clustering Trees

- Generalization of decision trees towards predicting structured outputs

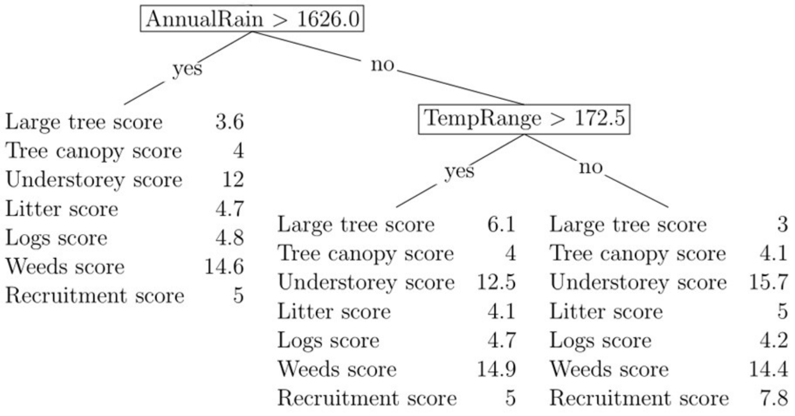
The top-down induction algorithm for PCTs



```
Procedure PCT
Input: A dataset $E$
Output: A predictive clustering tree
$(t^*, h^*, \mathcal{P}^*) = \mathrm{BestTest}(E)$
if $t^* \neq none$ then
    foreach $E_i \in \mathcal{P}^*$ do
        $tree_i = \mathrm{PCT}(E_i)$
    end
    return node($t^*$, $\bigcup_i\{tree_i\}$)
else
    return leaf(Prototype($E$))
end
```

```
Procedure BestTest
Input: A dataset $E$
Output: the best test ($t^*$), its heuristic score ($h^*$) and the partition ($\mathcal{P}^*$) it induces on the dataset ($E$)
$(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
foreach possible test $t$ do
    $\mathcal{P}$ = partition induced by $t$ on $E$
    $h = Var_f(E, Y) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var_f(E_i, Y)$
    if $(h > h^*) \wedge \mathrm{Acceptable}(t, \mathcal{P})$ then
        $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
    end
end
return $(t^*, h^*, \mathcal{P}^*)$
```

X is descriptive space, Y is target space, and E is a set of labeled examples



Variance function considers only target space Y

Semi-supervised predictive clustering trees

Variance function: Variance of **target** space **+** Variance of **descriptive** space

$$Var_f(E,Y,X) = \overbrace{w \cdot Var_f(E,Y)} + \overbrace{(1-w) \cdot Var_f(E,X)}$$

$\boldsymbol{w} \in [\mathbf{0}, \mathbf{1}]$ = controls the amount of supervision:

$$\begin{array}{ccc} w = 0 & 0 < w < 1 & w = 1 \\ \text{Unsupervised} & \text{Semi}-\text{supervised} & \text{Supervised} \end{array}$$

- Where X is descriptive space, Y is target space, and $E = E_l \cup E_u$ is a set of labeled and **unlabled examples**
- **Assumption:** examples similar in descriptive space have similar targets as well

Levatić, J., Kocev, D., Ceci, M., & Džeroski, S. (2018). Semi-supervised trees for multi-target regression. *Information Sciences*.

Levatić, J. (2017). *Semi-supervised Learning for Structred Output Prediction: Doctoral Dissertation* (Doctoral dissertation).

Semi-supervised predictive clustering trees

Variance of **target** space:

$$Var_f(E,Y) = \begin{cases} \frac{1}{T}\cdot\sum_{i=1}^{T} Var(E,Y_i),\ \text{if } Y \text{ consists of } T \text{ continuous variables} \\ \frac{1}{T}\cdot\sum_{i=1}^{T} Gini(E,Y_i),\ \text{if } Y \text{ consists of } T \text{ nominal variables} \\ \frac{1}{\sum\limits_{l=1}^{|C|} w(c_l)} \left(\frac{1}{|E|}\cdot\sum\limits_{e_i\in E_l} d(L_i,\overline{L})^2\right),\ \text{if } Y \text{ is a hierarchy of classes} \end{cases}$$

Can handle several structured output types

Variance of **descriptive** space:

$$Var_f(E,X) = \frac{1}{D}\cdot\left(\sum_{X_i \text{ is numeric}} Var(E,X_i) + \sum_{X_j \text{ is nominal}} Gini(E,X_j)\right)$$

Can handle numeric and nominal attributes

- Var and $Gini$ are normalized by variance on the entire training set

Levatić, J., Kocev, D., Ceci, M., & Džeroski, S. (2018). Semi-supervised trees for multi-target regression. *Information Sciences*.

Levatić, J. (2017). *Semi-supervised Learning for Structred Output Prediction: Doctoral Dissertation* (Doctoral dissertation).

Semi-supervised random forests

- Based on random forests for structured outputs (Kocev et al. 2013)
- Semi-supervised PCTs used as base learners

Statistical analysis

p-values of Wilcoxon paired signed rank test ($\alpha = 0.05$)*

Methods			Number of labeled examples				
			50	100	200	350	500
Multi-target regression							
PCT	vs.	SSL-PCT	0.093	**0.022**	**0.028**	**0.022**	**0.009**
RF	vs.	SSL-RF	0.959	0.445	0.445	0.333	0.445
Multi-label classification							
PCT	vs.	SSL-PCT	**0.013**	**0.008**	**0.008**	0.093	0.053
RF	vs.	SSL-RF	0.241	0.415	0.262	0.308	0.575
Hierarchical multi-label classification							
PCT	vs.	SSL-PCT	0.834	0.093	**0.028**	**0.028**	**0.028**
RF	vs.	SSL-RF	0.345	0.345	0.249	0.345	0.345

*In all tests, semi-supervised algorithms have better sum of ranks

SSL-PCTs for primitive outputs

p-values of Wilcoxon paired signed rank test ($\alpha = 0.05$)*

Methods			Number of labeled examples					
			25	50	100	200	350	500
Binary classification								
PCT	vs.	SSL-PCT	**0.009**	0.388	0.066	**0.005**	**0.019**	**0.019**
RF	vs.	SSL-RF	0.529	0.192	**0.002**	0.099	0.093	**0.012**
Multi-class classification								
PCT	vs.	SSL-PCT	0.248	0.084	**0.014**	**0.007**	0.192	0.081
RF	vs.	SSL-RF	0.563	**0.011**	**0.011**	**0.003**	**0.004**	**0.02**
Regression								
PCT	vs.	SSL-PCT	**0.011**	**0.01**	**0.004**	0.367	0.48	0.583
RF	vs.	SSL-RF	**0.008**	**0.065**	**0.008**	**0.023**	**0.034**	0.126

*In all tests, semi-supervised algorithms have better sum of ranks

Semi-supervised random forests

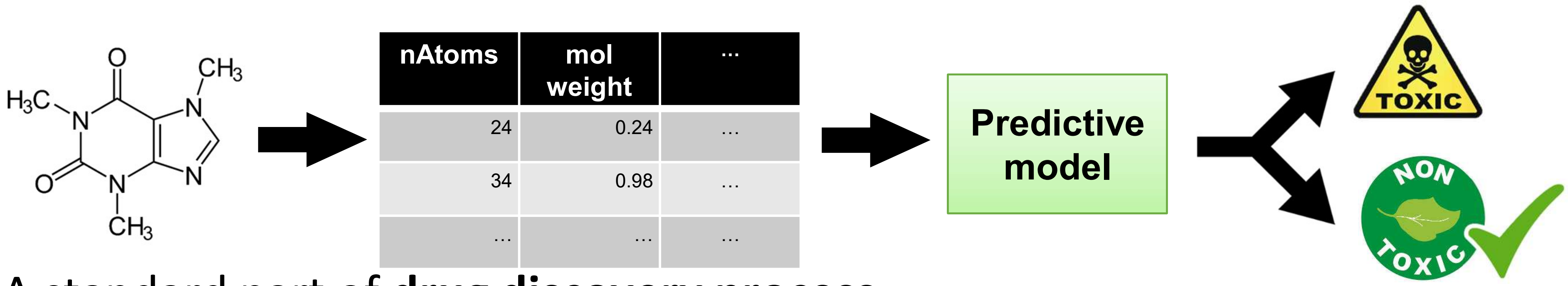
Self-training for multi-target regression (Levatić et al. 2017):

- Another semi-supervised method we developed based on random forests
- Iteratively uses its own predictions on unlabelled data as additional training examples

Levatić, J., Kocev, D., Ceci, M., & Džeroski, S. (2018). Semi-supervised trees for multi-target regression. *Information Sciences*.

Quantitative Structure-Activity Relationship (QSAR)

Predict biological activity of a molecule from its structure



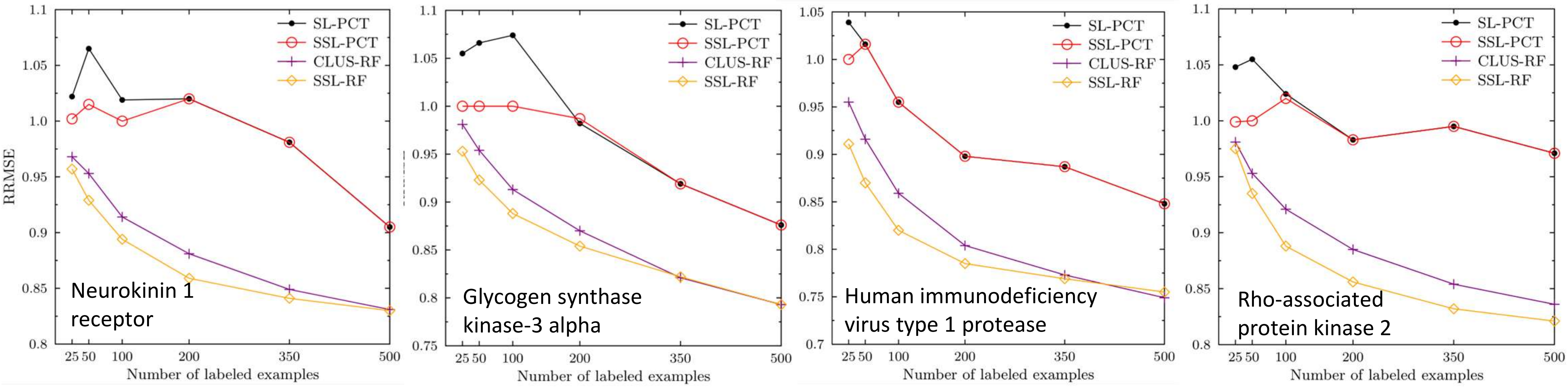
nAtoms	mol weight	…
24	0.24	…
34	0.98	…
…	…	…

Predictive model

A standard part of **drug discovery process**

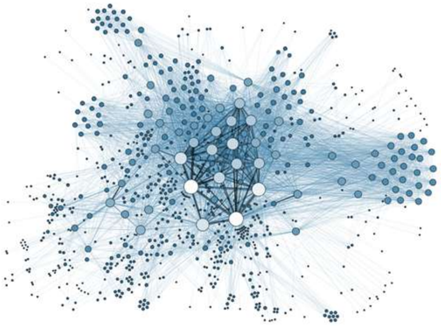
Quantitative Structure-Activity Relationship (QSAR)

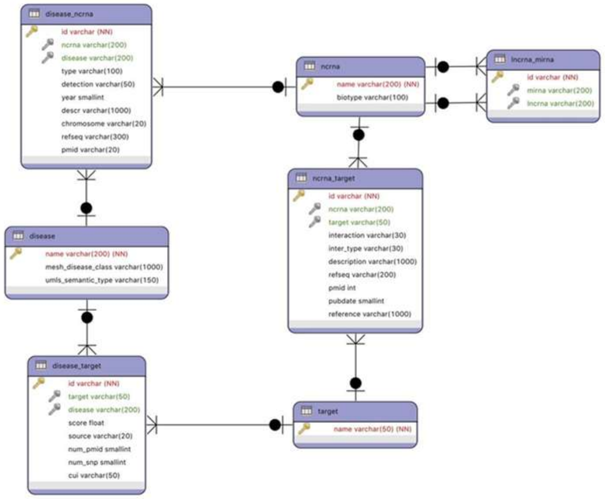
- Prediction of activity of **4 biological targets** from ChEMBL database
- Semi-supervised **regression trees** and random forests



Levatić, J., Ceci, M., Stepišnik, T., Džeroski, S., & Kocev, D. (2020). Semi-supervised regression trees with application to QSAR modelling. *Expert Systems with Applications*.

Analysis of Network and Relational Data





Autocorrelation

- Given a random variable Y representing the output of some observations x_i, and a distance function defined on observations, autocorrelation is the correlation among output values y_i strictly attributable to the proximity of observations according to the distance function.
- **Autocorrelation introduces a deviation from the independent observations' assumption of classical statistics.**
- Positive (negative) autocorrelation is the tendency for similar (dissimilar) values to cluster.
- Positive autocorrelation is more common than negative autocorrelation in spatial and social phenomena.

Positive autocorrelation vs smoothness assumption

In the **semi-supervised setting** (when the similarity between two observations is defined so that two observations are never considered similar when they are separated by low-density regions):

Positive autocorrelation entails the semi-supervised smoothness assumption

Positive autocorrelation vs smoothness assumption

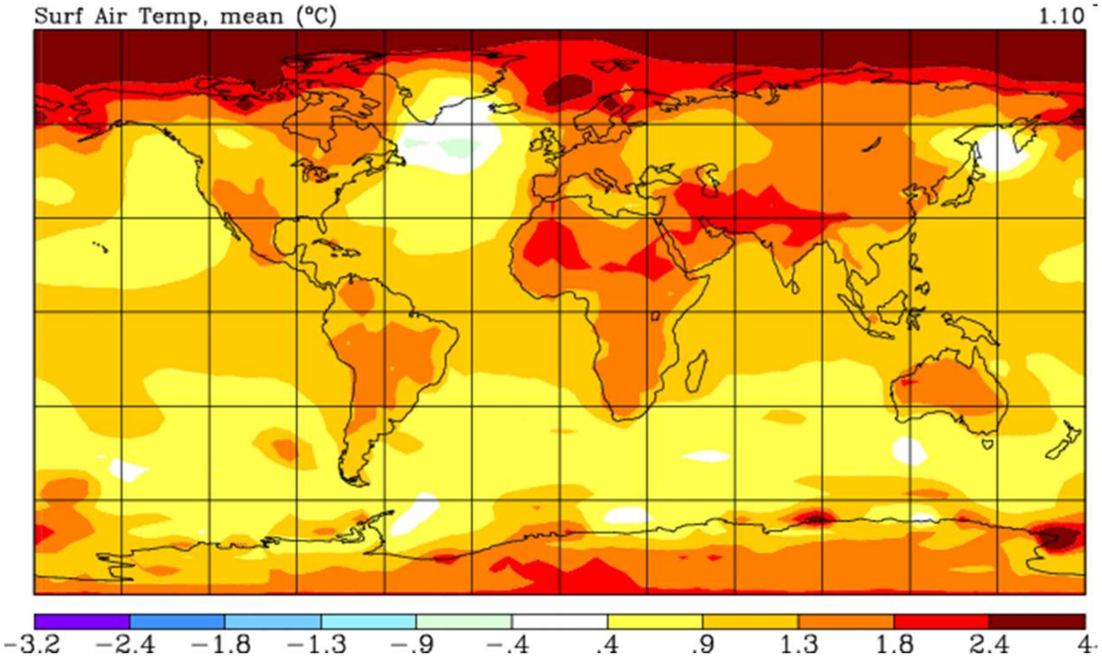
Autocorrelation is valid in networked data, in spatial data (spatial autocorrelation), in relational data:

- sociology (e.g., social relations affect social influence),
- web mining (e.g., related pages on the same topic),
- social networks (e.g. homophily property),
- bioinformatics (e.g., proteins located in the same place in a cell are more likely to share the same function than randomly selected proteins).

In these fields, the “distance” should reflect the properties of interest.

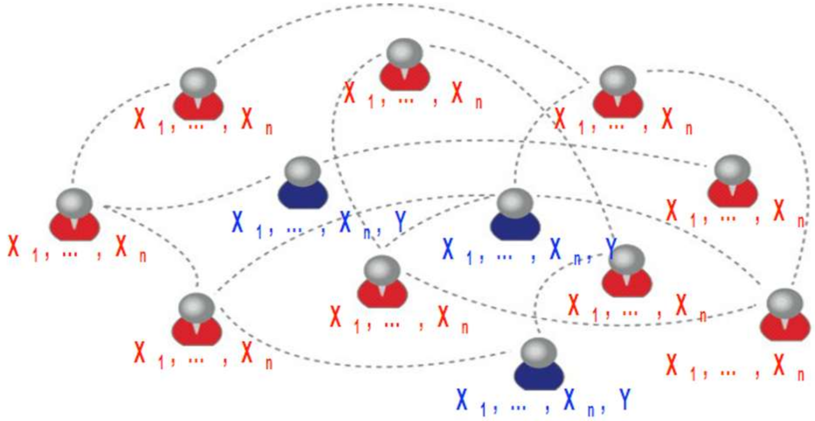
Spatial Data

Features tend to take values, for pairs of observations that are spatially close, that are more similar than expected for random pairs of observations.



Networked Data

- Nodes represent entities
- Links represent existing relations between entities
  - Nodes with known labels are interlinked with nodes for which the label is unknown
  - Labels are sparse



Examples:
- Internet
- Social networks
- Sensor networks ...

Networked Data

A collection of interconnected entities

Entities can be

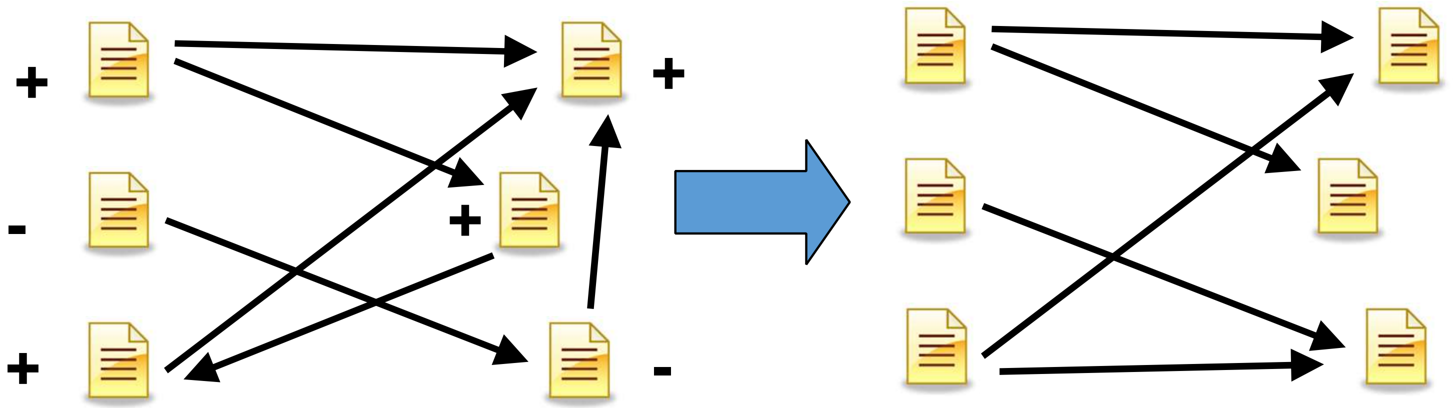
- homogeneous/heterogeneous
- Labelled/unlabelled
- Described by a single / multiple attribute(s) / structured representations
- Defined at various levels of abstractions

Connections/Links can be

- Homogeneous / heterogeneous
- Labelled / unlabelled
- Binary / n-ary
- Defined at various levels of abstraction

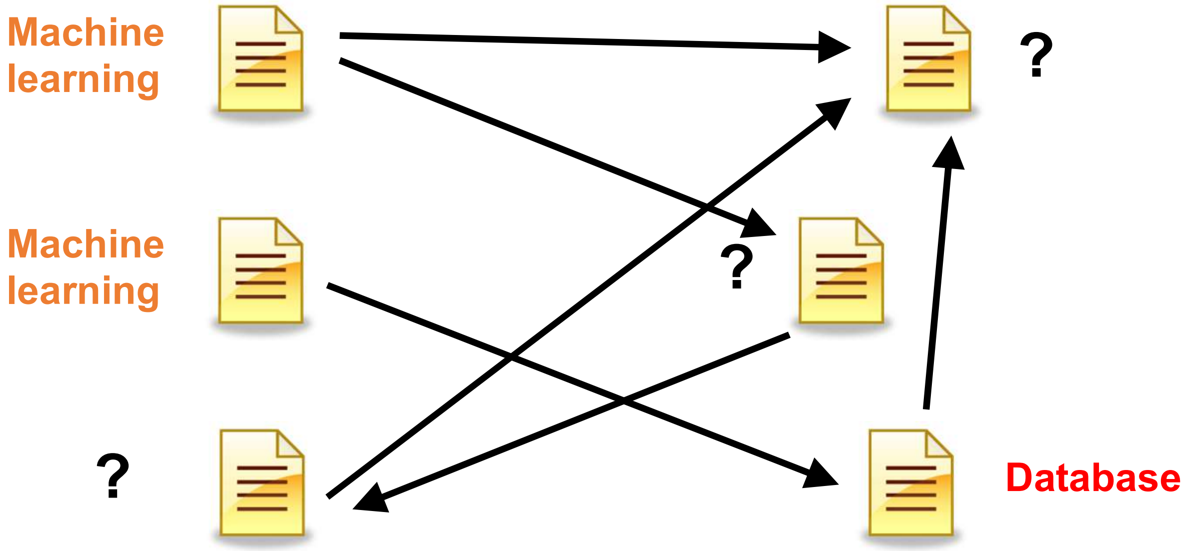
Across-Network Inference (inductive)

- Learning from one network and applying the learned model to a separate, presumably similar, network.



Within-Network Inference (transductive/semi-supervised)

Training entities are connected directly to those entities whose labels are to be estimated

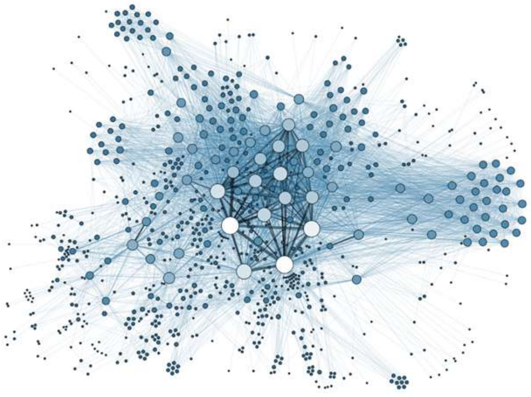


Biological Network Analysis

Semi-Supervised Multi-View Learning for Gene Network Reconstruction

We proposed a semi-supervised **multi-view learning method** to reconstruct the structure of gene regulatory networks from gene expression data. The proposed method:

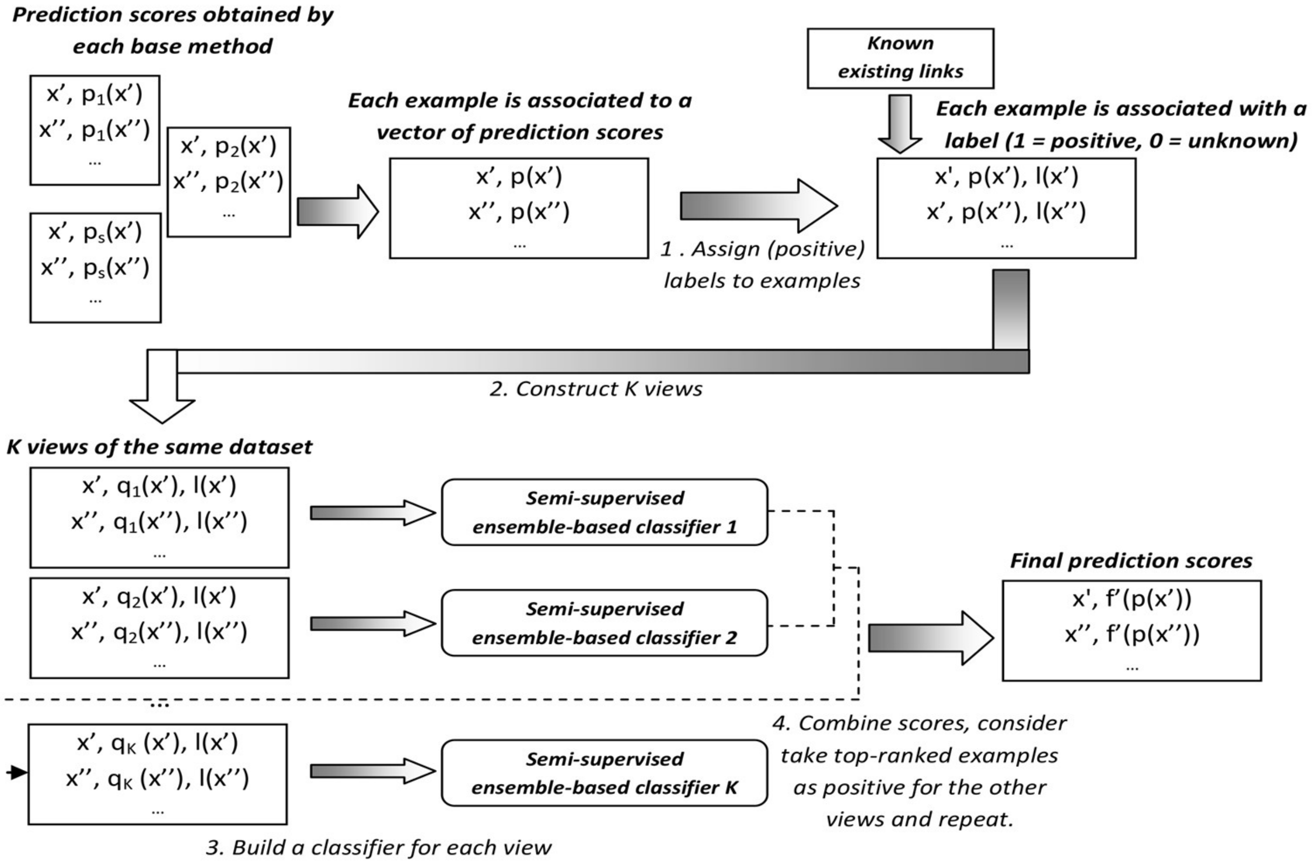
- **learns to combine the predictions** of multiple prediction methods
- is able to work in the **semi-supervised positive-unlabeled** setting, or in the **unsupervised setting**
- is able to manage a **high unbalancing** in the data
- **identifies *k* views** to build *k* classifiers, and exploits slight differences between multiple (possibly related/similar) prediction methods, avoiding issues due to collinearity



M. Ceci, G. Pio, V. Kuzmanovski, S. Dzeroski, *Semi-Supervised Multi-View Learning for Gene Network Reconstruction*, PLoS One 10(12): e0144031, 2015

Biological Network Analysis

Semi-Supervised Multi-View Learning for Gene Network Reconstruction



Prediction scores obtained by each base method

$x', p_1(x')$
$x'', p_1(x'')$
...

$x', p_2(x')$
$x'', p_2(x'')$
...

$x', p_s(x')$
$x'', p_s(x'')$
...

Each example is associated to a vector of prediction scores

$x', p(x')$
$x'', p(x'')$
...

1 . Assign (positive) labels to examples

Known existing links

Each example is associated with a label (1 = positive, 0 = unknown)

$x', p(x'), l(x')$
$x', p(x''), l(x'')$
...

2. Construct K views

K views of the same dataset

$x', q_1(x'), l(x')$
$x'', q_1(x''), l(x'')$
...

$x', q_2(x'), l(x')$
$x'', q_2(x''), l(x'')$
...

...

$x', q_K (x'), l(x')$
$x'', q_K (x''), l(x'')$
...

Semi-supervised ensemble-based classifier 1

Semi-supervised ensemble-based classifier 2

Semi-supervised ensemble-based classifier K

3. Build a classifier for each view

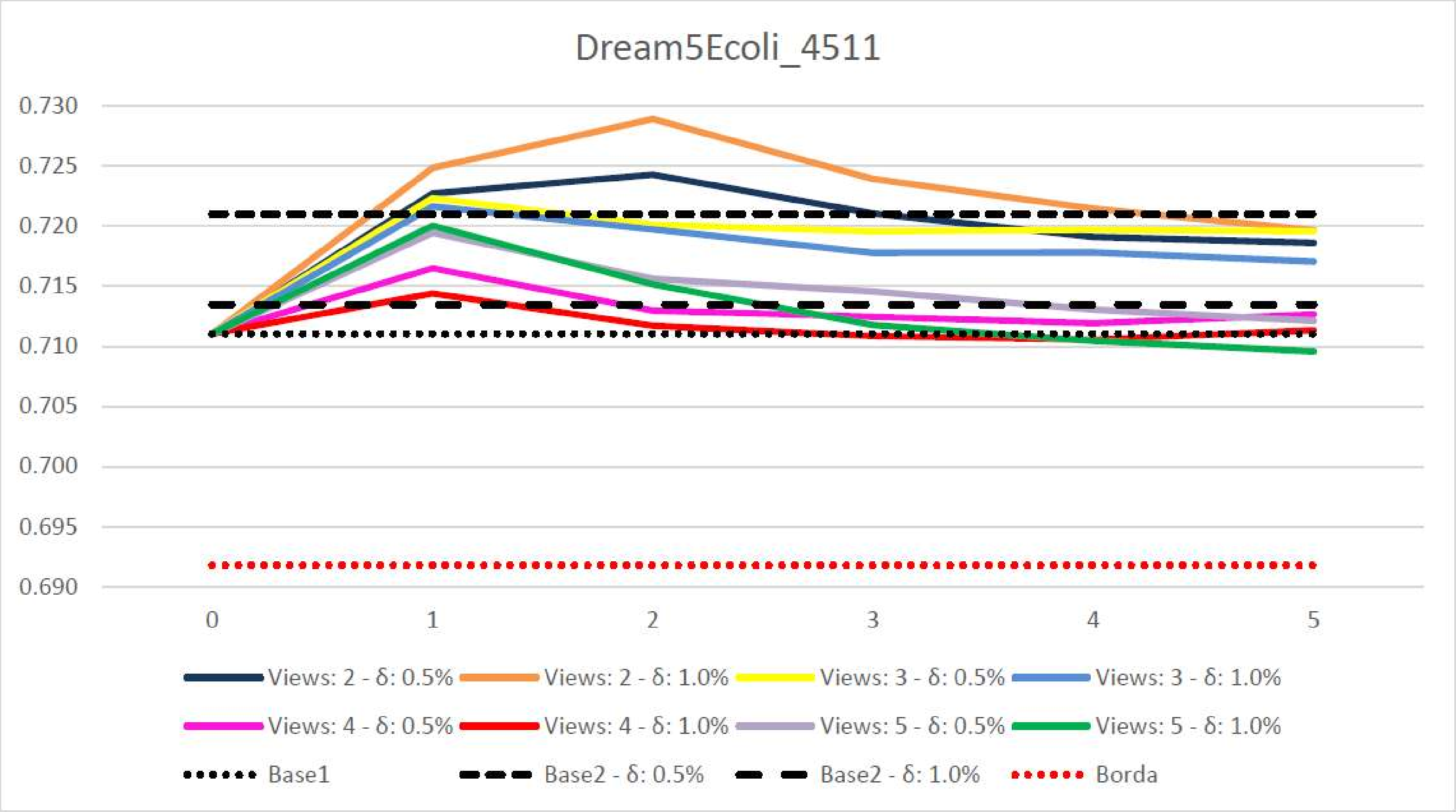
4. Combine scores, consider take top-ranked examples as positive for the other views and repeat.

Final prediction scores

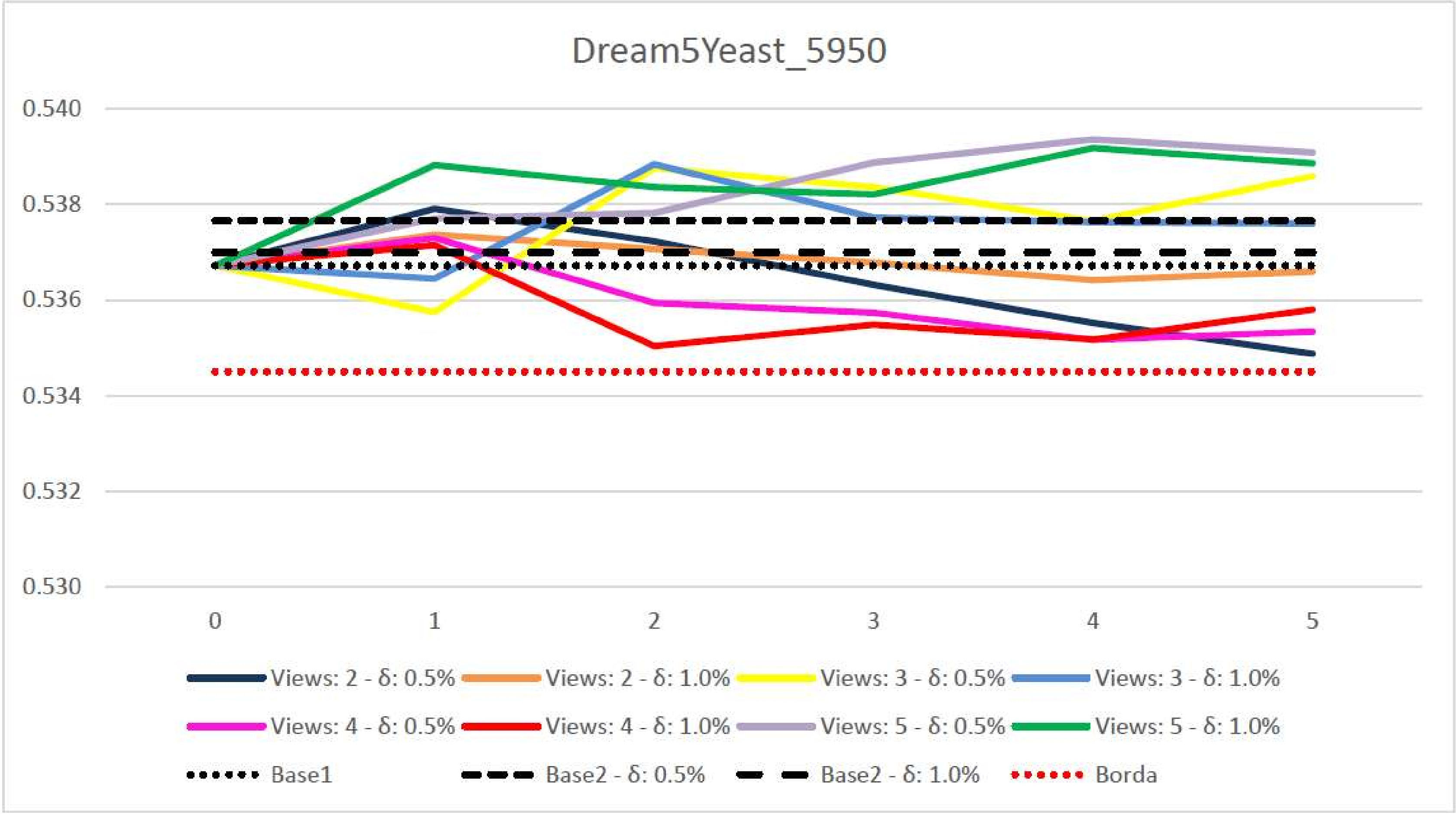
$x', f'(p(x'))$
$x'', f'(p(x''))$
...

M. Ceci, G. Pio, V. Kuzmanovski, S. Dzeroski, *Semi-Supervised Multi-View Learning for Gene Network Reconstruction*, PLoS One 10(12): e0144031, 2015

Some experimental results

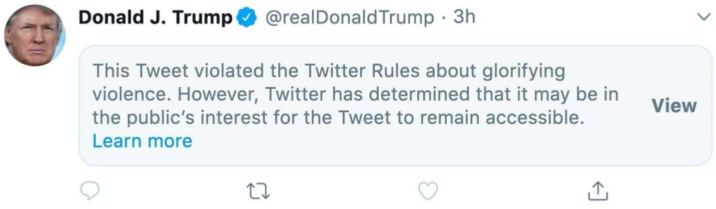


Some experimental results



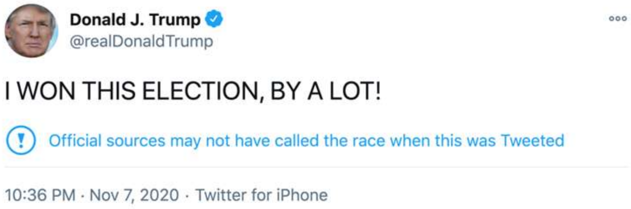
Social media: **the problem**

- Social media can be **harmful** since they can be exploited by **risky users** to harass people or influence them to perform illegal acts.
- Everyday, many social pages spread religious fundamentalism and political extremism.



Donald J. Trump @realDonaldTrump · 3h

This Tweet violated the Twitter Rules about glorifying violence. However, Twitter has determined that it may be in the public's interest for the Tweet to remain accessible.
Learn more



Donald J. Trump @realDonaldTrump

I WON THIS ELECTION, BY A LOT!

Official sources may not have called the race when this was Tweeted

10:36 PM · Nov 7, 2020 · Twitter for iPhone

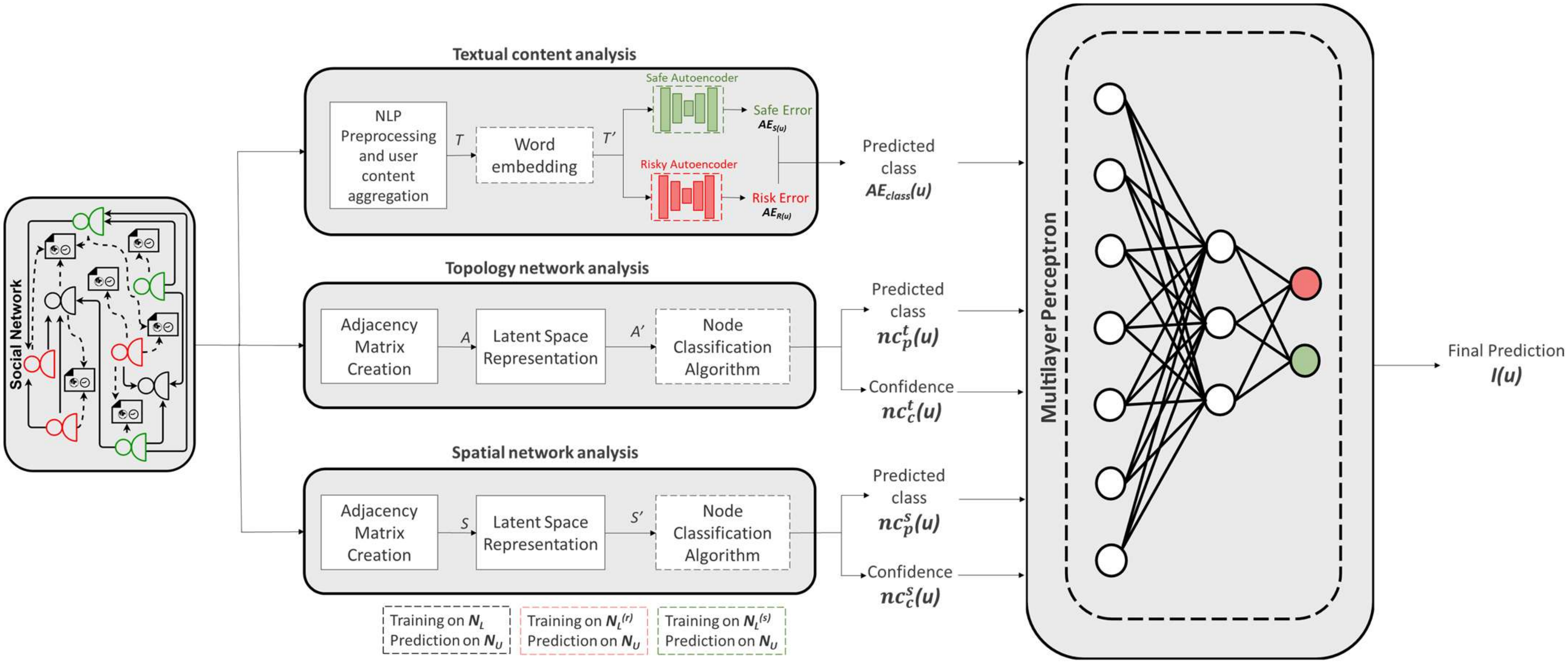
SAIRUS framework: **idea**

Classical social network analysis frameworks consider only **one perspective** when classifying users:

- the network topology – i.e., the relationships among users in the network (follows, likes, etc…),
- the user generated content – i.e., the posts or the tweets shared by a specific user.

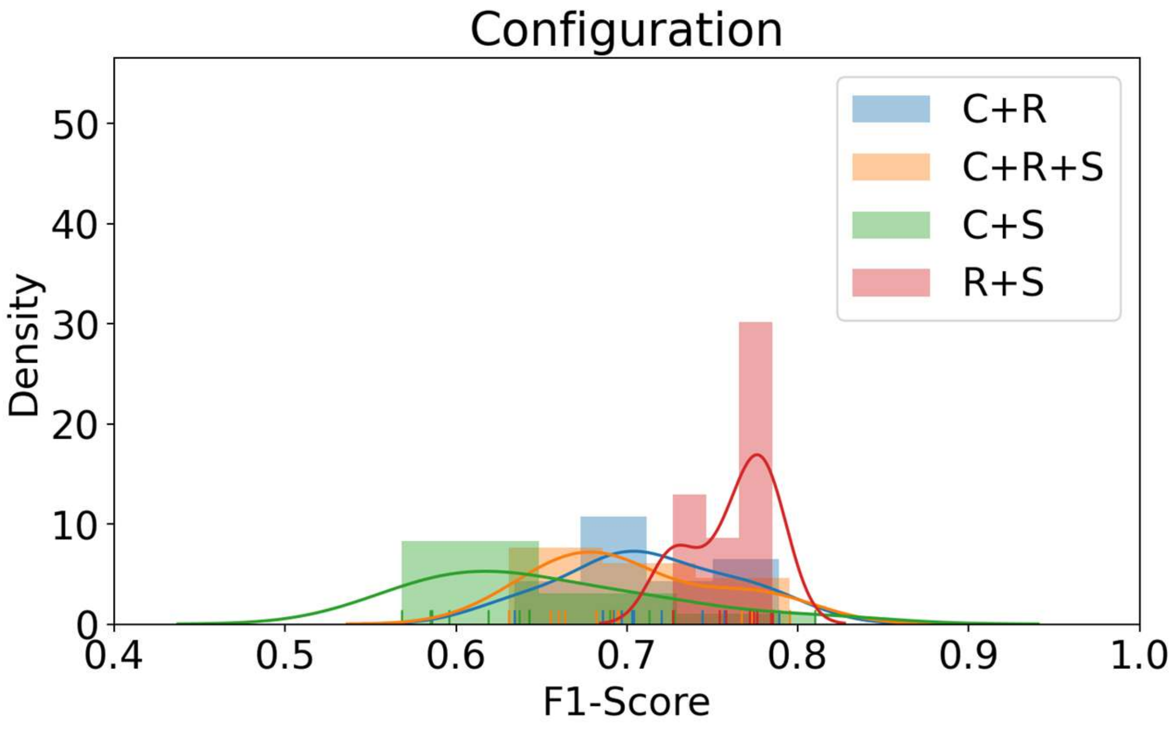
Our system not only aims to **exploit both aspects**, but also considers **spatial information**.

SAIRUS framework: **a general view**


Social Network
Textual content analysis
NLP Preprocessing and user content aggregation
T
Word embedding
T'
Safe Autoencoder
Safe Error
$AE_{S(u)}$
Risky Autoencoder
Risk Error
$AE_{R(u)}$
Predicted class
$AE_{class}(u)$
Topology network analysis
Adjacency Matrix Creation
A
Latent Space Representation
A'
Node Classification Algorithm
Predicted class
$nc_p^t(u)$
Confidence
$nc_c^t(u)$
Spatial network analysis
Adjacency Matrix Creation
S
Latent Space Representation
S'
Node Classification Algorithm
Predicted class
$nc_p^s(u)$
Confidence
$nc_c^s(u)$
Training on N_L Prediction on N_U
Training on $N_L^{(r)}$ Prediction on N_U
Training on $N_L^{(s)}$ Prediction on N_U
Multilayer Perceptron
Final Prediction
$I(u)$

Pellicani, A., Pio, G., Redavid, D., & Ceci, M. (2023). SAIRUS: Spatially-aware identification of risky users in social networks. In Information Fusion (Vol. 92, pp. 435–449). Elsevier

Result Comparison: **System Configuration**



C: content

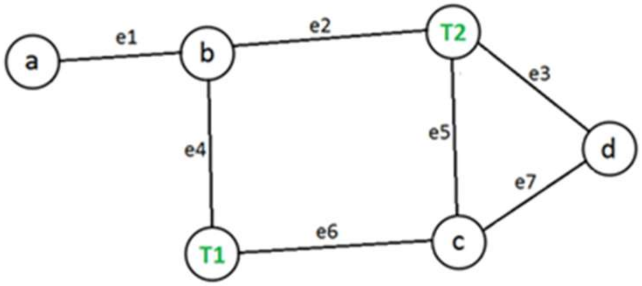
R: relationships

S: spatial information

Heterogeneous Network Analysis

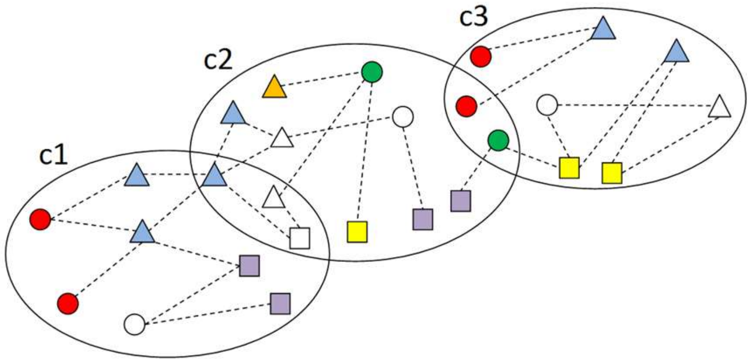
Multi-type clustering and classification from heterogeneous networks

Contribution: a novel clustering algorithm that identifies **heterogeneous** (i.e., consisting of multiple types of objects and links), **overlapping** and **hierarchically organized clusters** from attributed heterogeneous networks, that are exploited also for **predictive purposes.**



The construction of the clusters is based on the concept of **meta-paths**, that are automatically identified from the network, and on the attributes of the nodes involved in the meta-paths.

Node classification and link prediction tasks are solved using a weighted majority voting approach, where the weight is based on the number of labelled examples in the clusters the considered unlabelled example falls into.



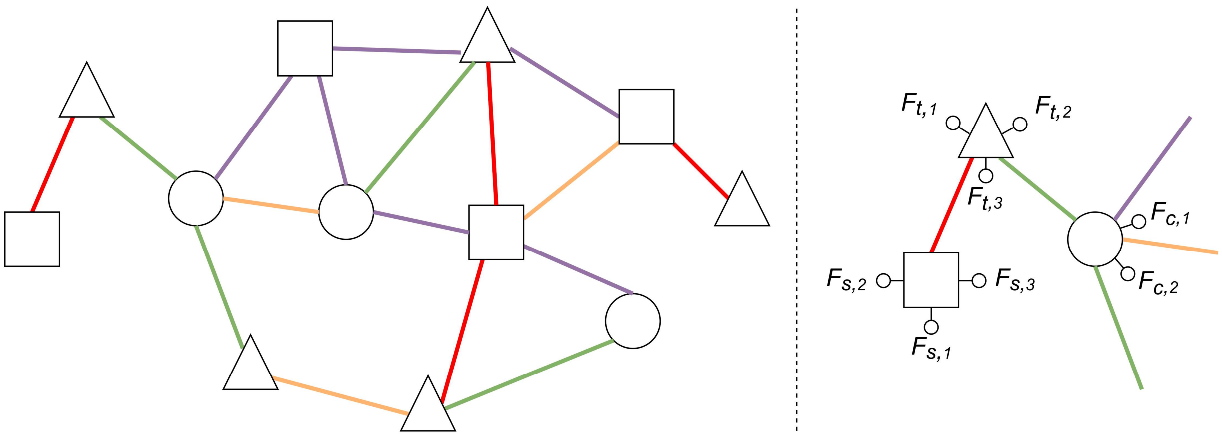
G. Pio, F. Serafino, D. Malerba, M. Ceci, *Multi-type clustering and classification from heterogeneous networks*, Information Sciences, 425:107-126, 2018

E. Barracchia, G.Pio, D. D'Elia, M. Ceci, *Prediction of new associations between ncRNAs and diseases exploiting multi-type hierarchical clustering*, BMC Bioinformatics 21, 70, 2020

LP-HCLUS

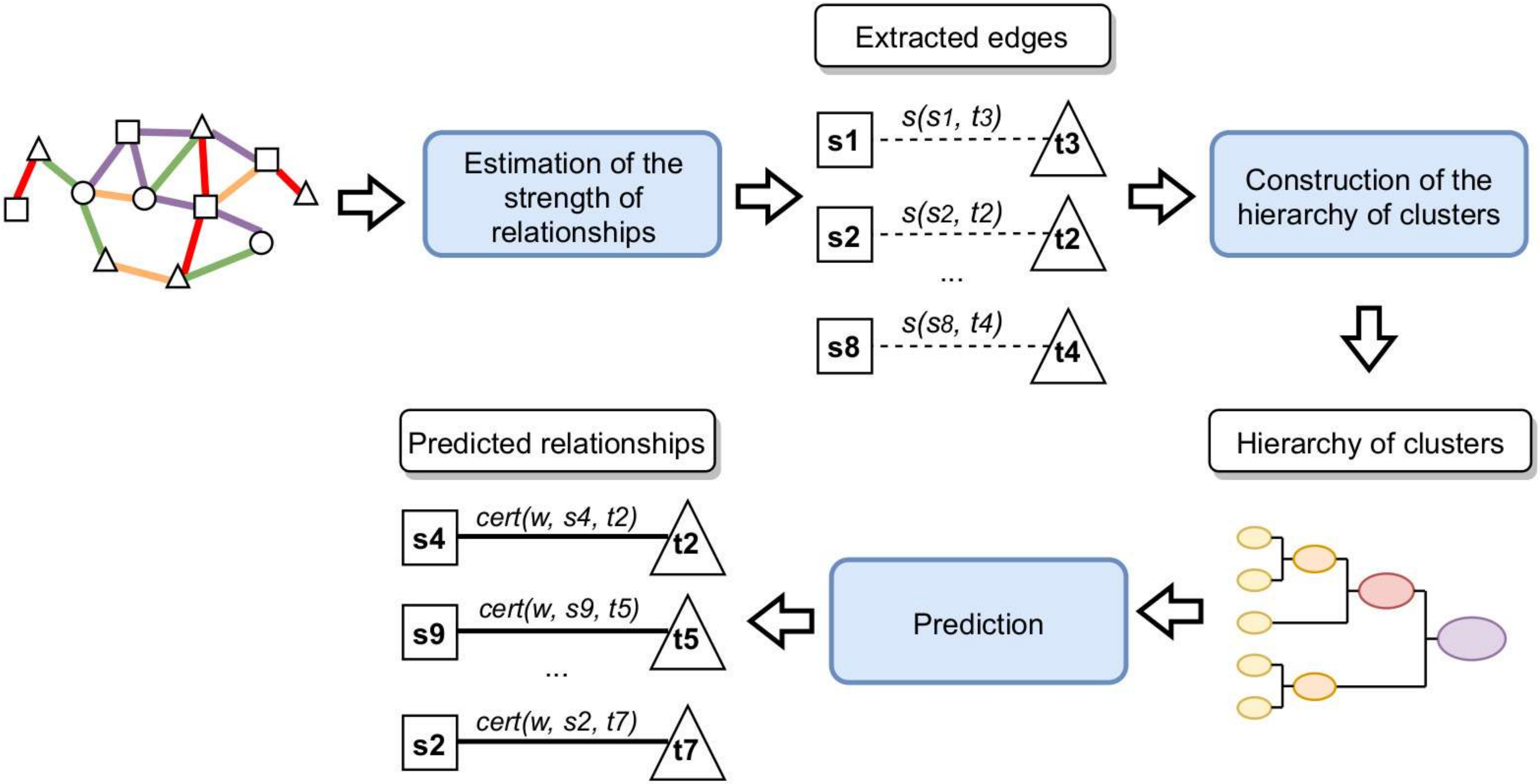
LP-HCLUS (Link Prediction through Hierarchical CLUStering):

- performs **link prediction on heterogeneous attributed networks**
- exploits a heterogeneous **clustering technique**
- adopts a **similarity measure** based on the **features** and the **relationships** in the network
- has been applied to the **biological domain**



LP-HCLUS

Workflow



Estimation of the strength of relationships

Extracted edges

Construction of the hierarchy of clusters

Hierarchy of clusters

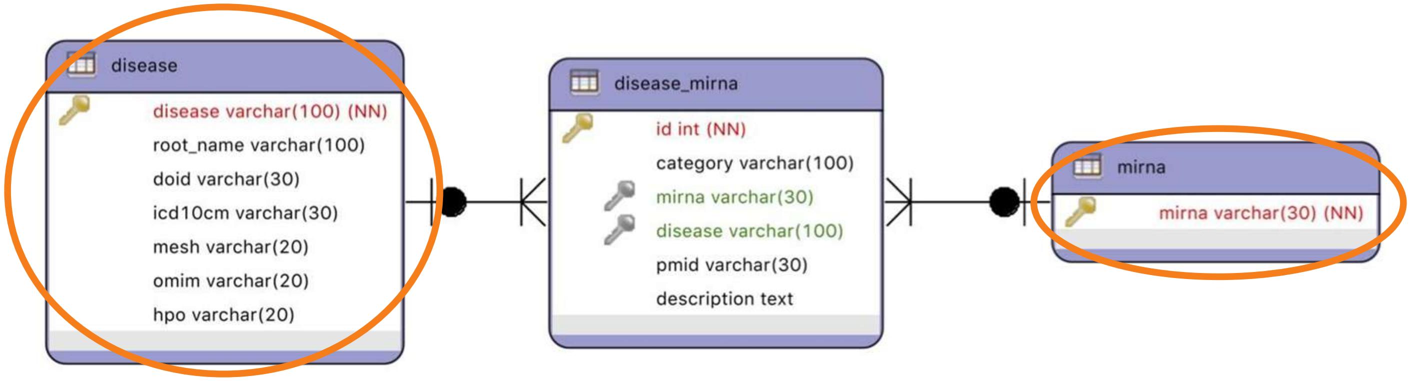
Prediction

Predicted relationships

LP-HCLUS

Quantitative evaluation

HMDD v3



disease
- disease varchar(100) (NN)
- root_name varchar(100)
- doid varchar(30)
- icd10cm varchar(30)
- mesh varchar(20)
- omim varchar(20)
- hpo varchar(20)

disease_mirna
- id int (NN)
- category varchar(100)
- mirna varchar(30)
- disease varchar(100)
- pmid varchar(30)
- description text

mirna
- mirna varchar(30) (NN)

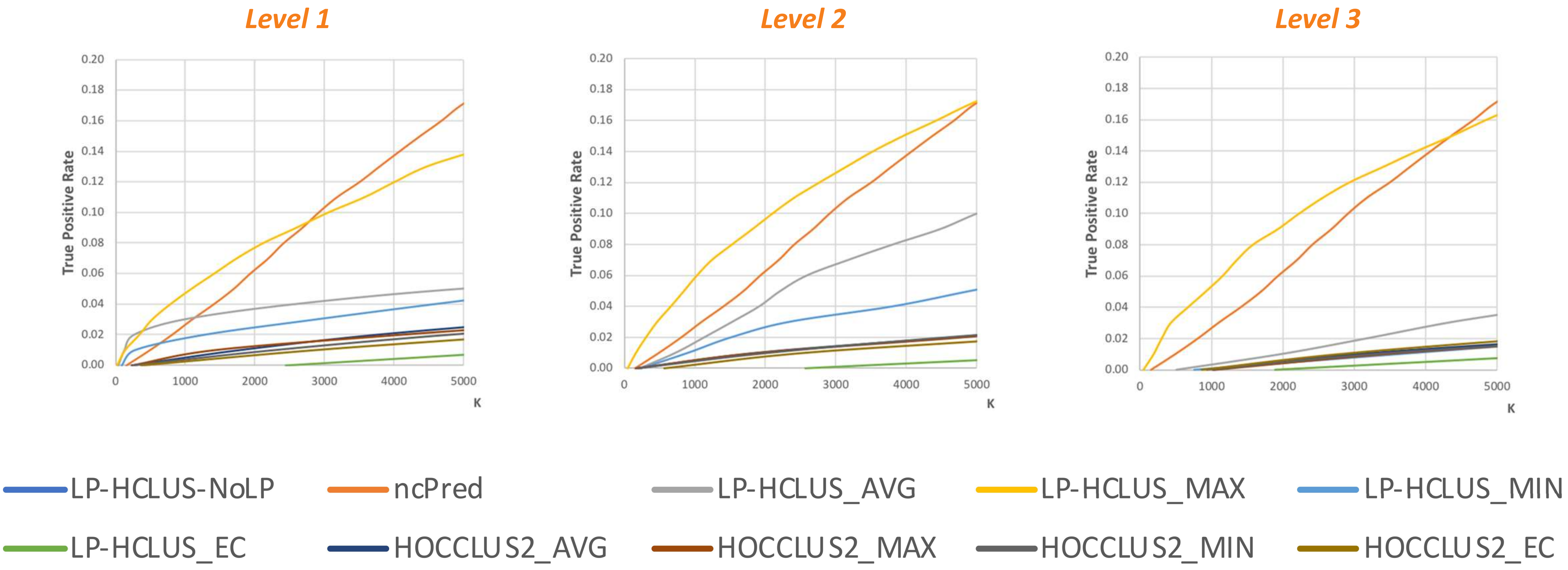
Table	# of instances
Disease	675
MiRNA	985
Disease - MiRNA	20,859

Dataset available at:
http://www.cuilab.cn/hmdd

LP-HCLUS

Quantitative evaluation

HMDD v3



Level 1

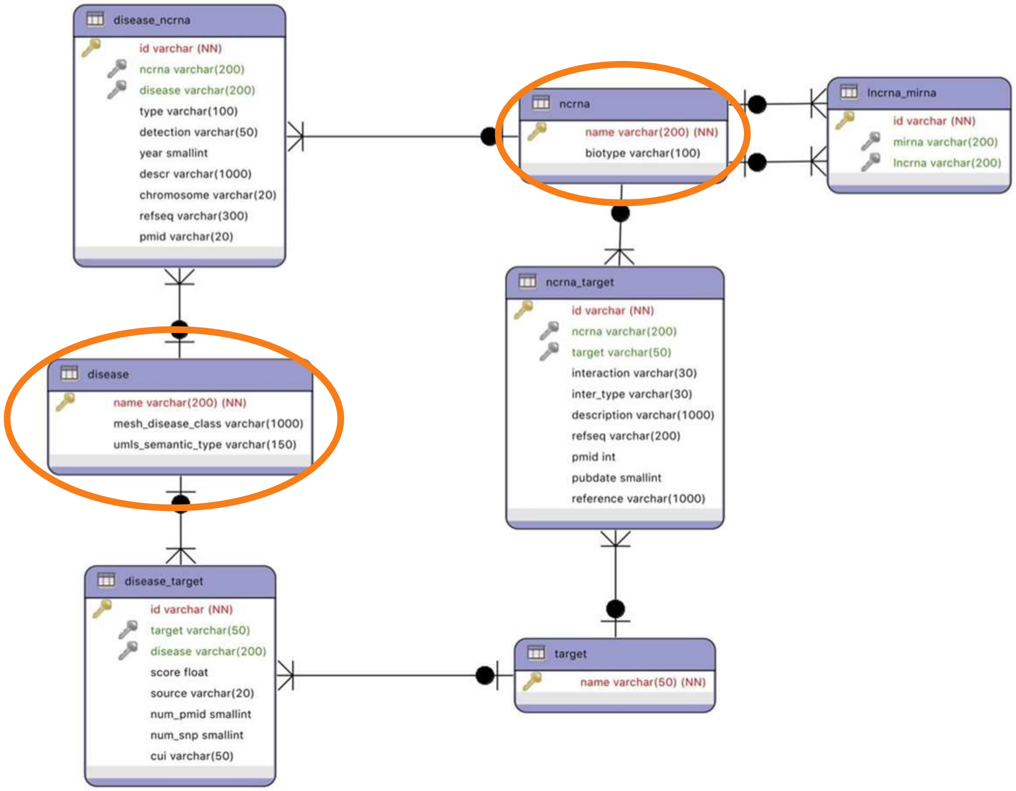
Level 2

Level 3

LP-HCLUS

Quantitative evaluation

Integrated Dataset



disease_ncrna
id varchar (NN)
ncrna varchar(200)
disease varchar(200)
type varchar(100)
detection varchar(50)
year smallint
descr varchar(1000)
chromosome varchar(20)
refseq varchar(300)
pmid varchar(20)

ncrna
name varchar(200) (NN)
biotype varchar(100)

lncrna_mirna
id varchar (NN)
mirna varchar(200)
lncrna varchar(200)

disease
name varchar(200) (NN)
mesh_disease_class varchar(1000)
umls_semantic_type varchar(150)

ncrna_target
id varchar (NN)
ncrna varchar(200)
target varchar(50)
interaction varchar(30)
inter_type varchar(30)
description varchar(1000)
refseq varchar(200)
pmid int
pubdate smallint
reference varchar(1000)

disease_target
id varchar (NN)
target varchar(50)
disease varchar(200)
score float
source varchar(20)
num_pmid smallint
num_snp smallint
cui varchar(50)

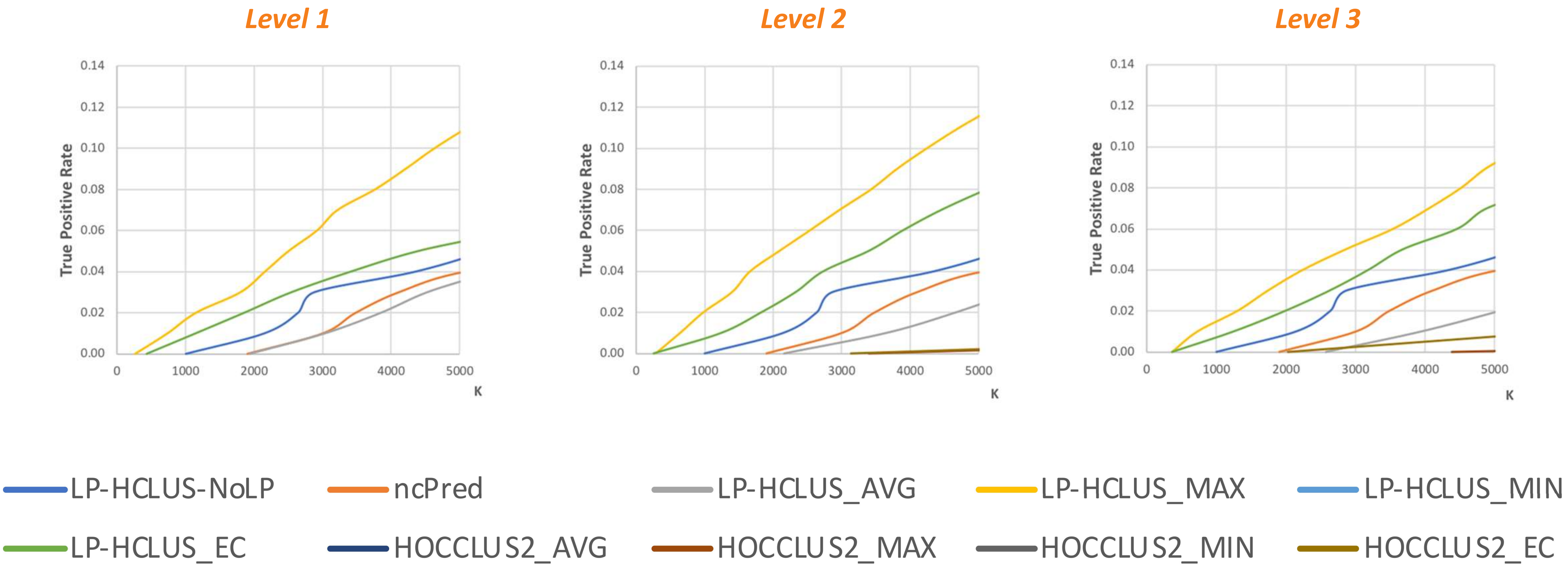
target
name varchar(50) (NN)

Table	# of instances
Disease	7,049
NcRNA	1,015
Target	90,242
Disease - NcRNA	3,830
Disease - Target	26,522
NcRNA - Target	1,055
LncRNA - MiRNA	70

LP-HCLUS

Quantitative evaluation

Integrated Dataset



Level 1

Level 2

Level 3

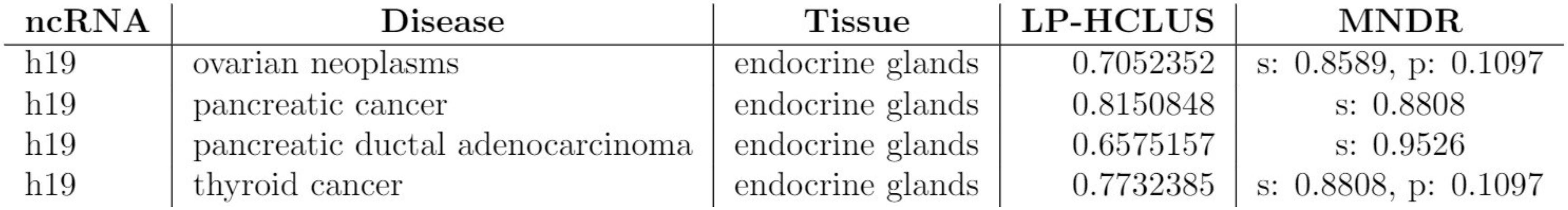
LP-HCLUS

Qualitative evaluation

In literature, the **lncRNA h19** appears in the regulation of many processes impacting diseases, but associations with **"bone diseases"**, as predicted by LP-HCLUS, are not reported.

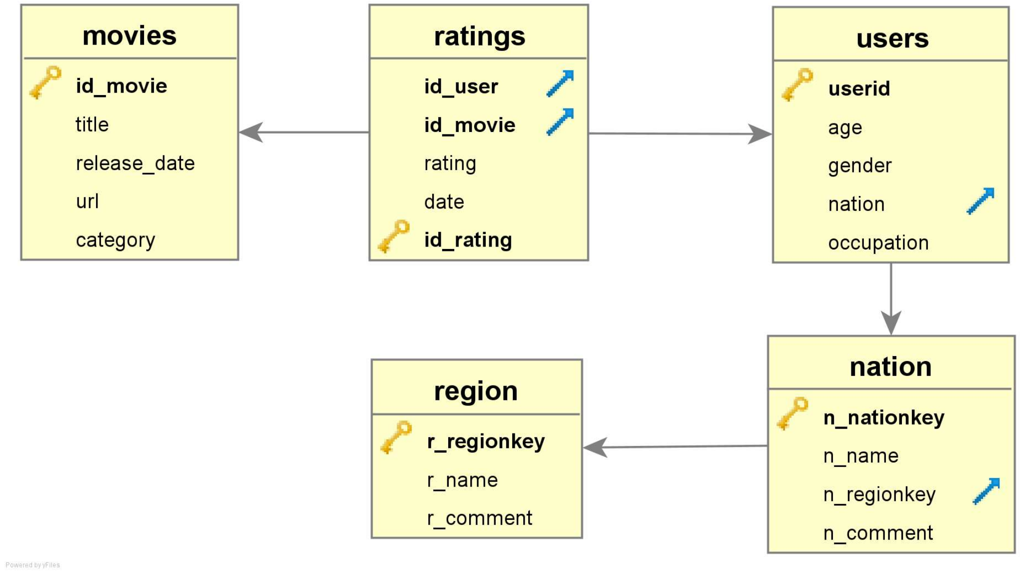
Bone diseases can have different origins and can be also related to **hyperfunction or hypofunction of the endocrine glands**. Both the output of LP-HCLUS and data in MNDR confirm the existence of associations between **h19** and diseases which involve **endocrine glands.**

This indicates that **h19** can have a relationship with **endocrine glands functions** and, therefore, can be related to **bone diseases** as predicted by LP-HCLUS.



ncRNA	Disease	Tissue	LP-HCLUS	MNDR
h19	ovarian neoplasms	endocrine glands	0.7052352	s: 0.8589, p: 0.1097
h19	pancreatic cancer	endocrine glands	0.8150848	s: 0.8808
h19	pancreatic ductal adenocarcinoma	endocrine glands	0.6575157	s: 0.9526
h19	thyroid cancer	endocrine glands	0.7732385	s: 0.8808, p: 0.1097

Relational Data



movies
id_movie
title
release_date
url
category

ratings
id_user
id_movie
rating
date
id_rating

users
userid
age
gender
nation
occupation

nation
n_nationkey
n_name
n_regionkey
n_comment

region
r_regionkey
r_name
r_comment

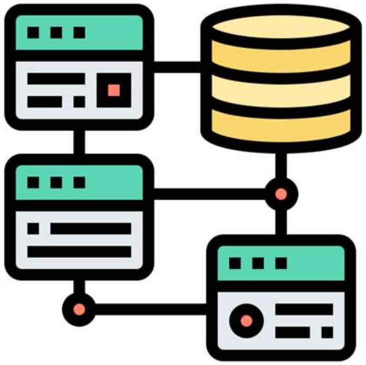
- Data stored in **multiple interconnected tables**
- Consider features of non-target tables *following the relations that connect tables*
- Handle complex relationships

  e.g. one movie can have many ratings from different users

Re3py

Re3py: A novel relational tree-based method

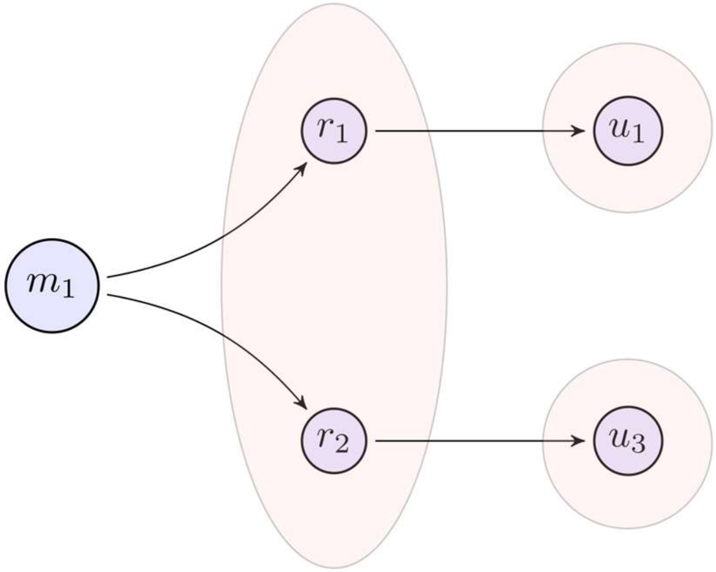
- Extends traditional tree ensembles to handle **relational data**
- **Structural approach** which preserves the original data structure and navigates the relational links directly during the learning process
- Split candidates are based on conditions across paths involving multiple tables and aggregates of attributes.
- Provides feature rankings in the relational context



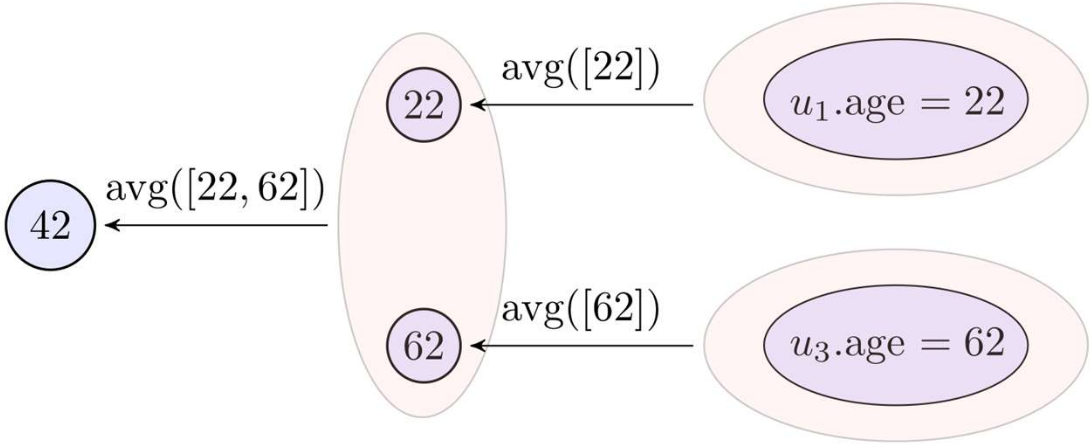
M.Petković, M.Ceci, G.Pio, B.Škrlj, K.Kersting, and S. Džeroski. Relational tree ensembles and feature rankings. Knowledge-Based Systems, 251:109254, 2022

Feature construction

1. Finding task-relevant objects for the movie m_1:
e.g. users who rated the movie



2. Aggregating the values:
e.g. average age of users who rated the movie



Semi-supervised Re3py

Working in the semi-supervised learning setting

- Extending the heuristics used during the tree construction (Gini) to also consider the descriptive space

$$Gini_f(E) = \underbrace{wGini_f^Y(E)}_{\textbf{Heuristics of the original Re3py}} + (1 - w)Gini_f^X(E)$$

where w ∈ [0, 1] controls how much the target space and the descriptive space contribute to the Gini estimation.

Gini over the descriptive space

$$Gini_f^X(E) = \frac{1}{D}\left(\sum_{X_i \in X \ and \ X_i \ is \ numeric} Var_i(E) \ + \sum_{X_i \in X \ and \ X_i \ is \ nominal} Gini_i(E)\right)$$

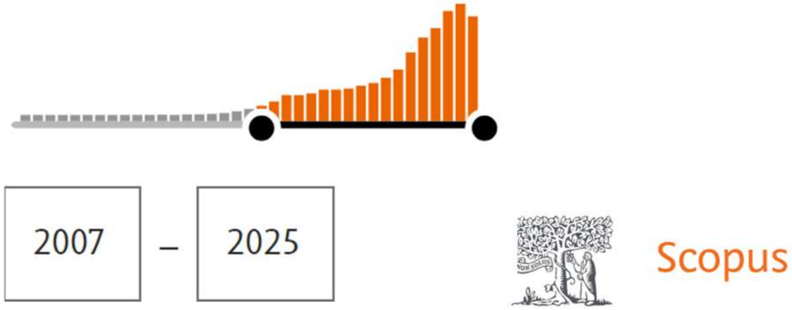
Experimental Setting

Dataset: Carcinogenesis

		min_sample_leaf = 1			min_sample_leaf = 5		
Method	**w**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Supervised	-	0.538	0.526	0.500	0.531	0.504	0.389
Semi-supervised	0.0	0.420	0.419	0.419	0.414	0.413	0.409
Semi-supervised	0.1	0.444	0.444	0.439	0.414	0.413	0.409
Semi-supervised	0.2	0.461	0.461	0.455	0.414	0.413	0.409
Semi-supervised	0.3	0.428	0.427	0.424	0.414	0.413	0.409
Semi-supervised	0.4	0.482	0.483	0.480	0.414	0.413	0.409
Semi-supervised	0.5	0.441	0.440	0.438	0.414	0.413	0.409
Semi-supervised	0.6	0.342	0.341	0.341	0.444	0.444	**0.439**
Semi-supervised	0.7	**0.590**	**0.588**	**0.575**	0.444	0.444	**0.439**
Semi-supervised	0.8	0.508	0.508	**0.508**	0.444	0.444	**0.439**
Semi-supervised	0.9	0.465	0.472	0.455	0.420	0.428	0.402
Semi-supervised	1.0	0.498	0.498	0.497	0.420	0.428	0.402

Conclusions

Very high interest in semi-supervised learning in the last 4-5 years



However, when analyzing scientific data new challenges arise and more work is necessary:

- Structured output prediction
- Network data
- Relational Data

Conclusions

Future work:

- Time series data
- Network data + Structured output prediction (e.g. gene function prediction)
- Network data + time series data (e.g. ecological data)
- …

Theoretical questions:

- Why many studies report of negative effects in Semi-supervised learning?
- How much the smoothness assumption influences the beneficial effects of Semi-supervised learning?

Thank you

Contact: michelangelo.ceci@uniba.it

This is the work of many people, including: Jurica Levatić, Gianvito Pio, Saso Dzeroski, Dragi Kocev, Donato Malerba, Antonio Pellicani, Emanuele Pio Barracchia, Vladimir Kuzmanovski, Annunziata D’Aversa, Francesco Serafino, Domenica D’Elia, Tomas Stepišnik, Domenico Redavid