Advanced Techniques for Mining Structured Data:

Graph Mining

Node prediction

Dr C.Loglisci

PhD Course in Computer Science and Mathematics XXXII cycle

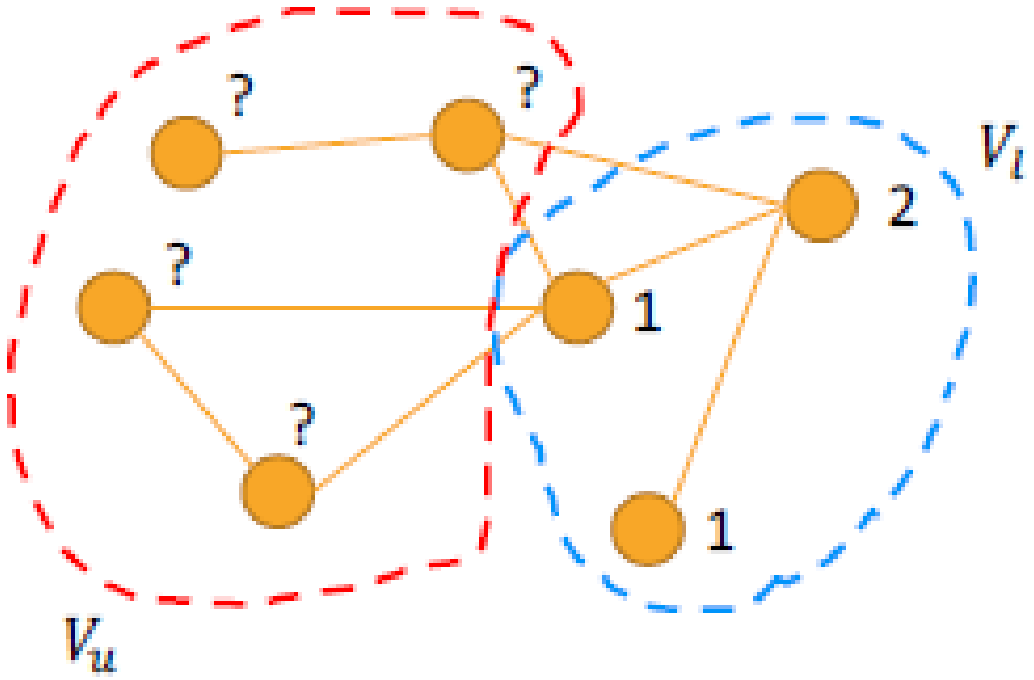
Motivations

- Not all the nodes have labels (nodes may be uncompleted due to different reasons, e.g., generated with misses)
- Labels provided by the users can be misleading
- Labels are sparse (some categories might be missing or incomplete)

- Suggesting new connections or contacts
- Automatically understand roles in a network (hubs, activators, influencing nodes, …)
- Study of diseases and cures
- Identify unusual behaviors or behavioral changes

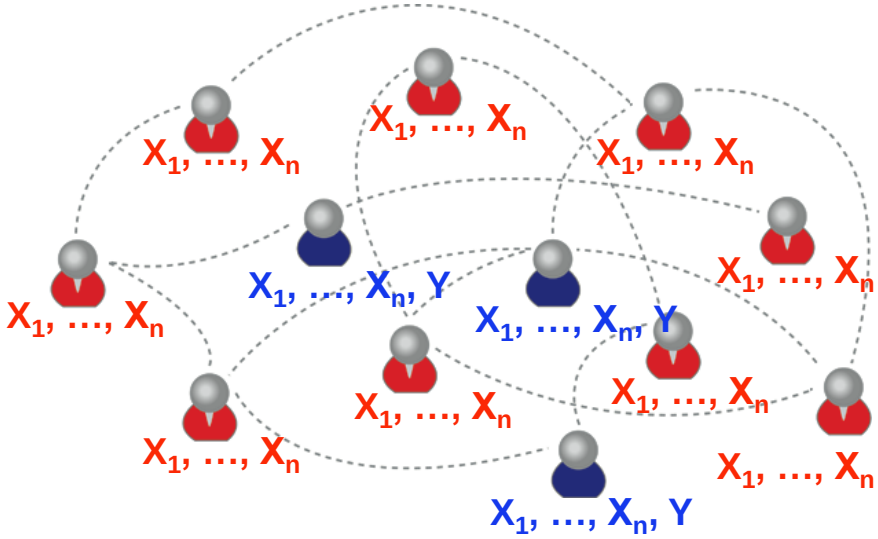
A Prediction problem

- Graph G: V, E,W with vertices V, edges E, weight matrix W
- Labeled nodes Vl $\subset V$, unlabeled nodes Vu= $V \setminus V$l



A Prediction problem

- Graph G: V, E,W with vertices V, edges E, weight matrix W
- Labeled nodes Vl $\subset V$, unlabeled nodes Vu= $V \setminus V$l
- Node described by an attribute set X:X1,…,Xm,Y (Xi independent, Y dependent)
- The goal is to estimate the attribute Y for each unlabeled node.



Importance of the network structure

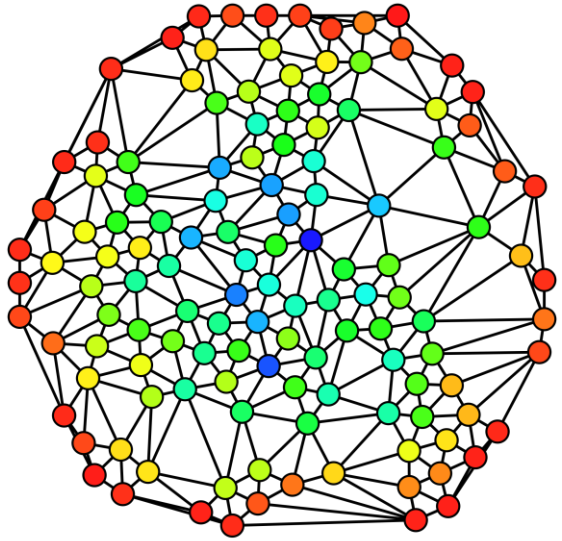
- The graph structure encodes important information for node prediction
- Two important concepts from social sciences:
  - Homophily: similar individuals are connected with similar people (friends of friends can be easily friends)
  - Co-citation regularity: if two people share a link most probably are similar in other connections (e.g., music tastes)
- So, it is reasonable to think that labels propagate in the network following the links, or the labels of nodes can influence the labels of linked nodes.

State-of-Art Approaches

- Similarity-based
  - find nodes that share the same characteristics with other nodes
- Iterative Convergence
  - learn a set of labels and propagate the information to similar nodes
- Label propagation
  - labeled nodes propagate the information to the neighbors with some probability

Issues

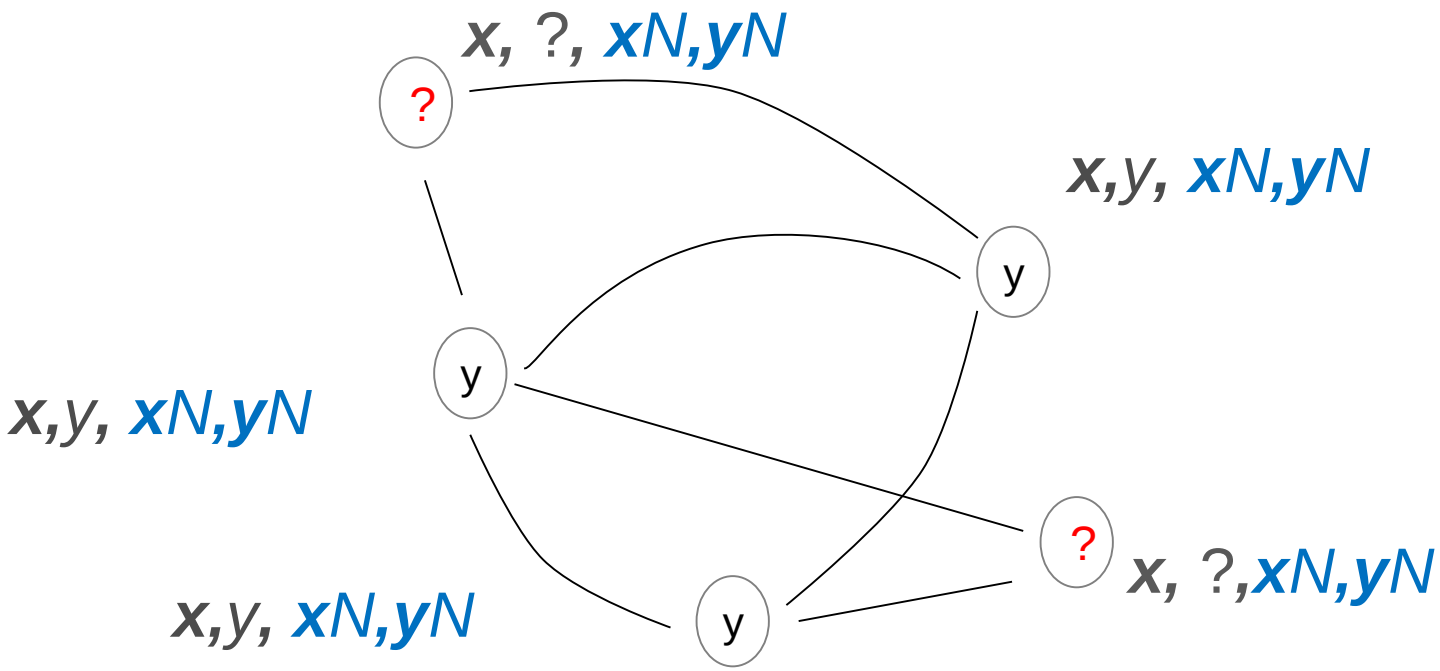
- Network auto-correlation:
  - The value of some attribute dependent by the value on the linked nodes.
  - positive and negative auto-correlation



Iterative Convergence

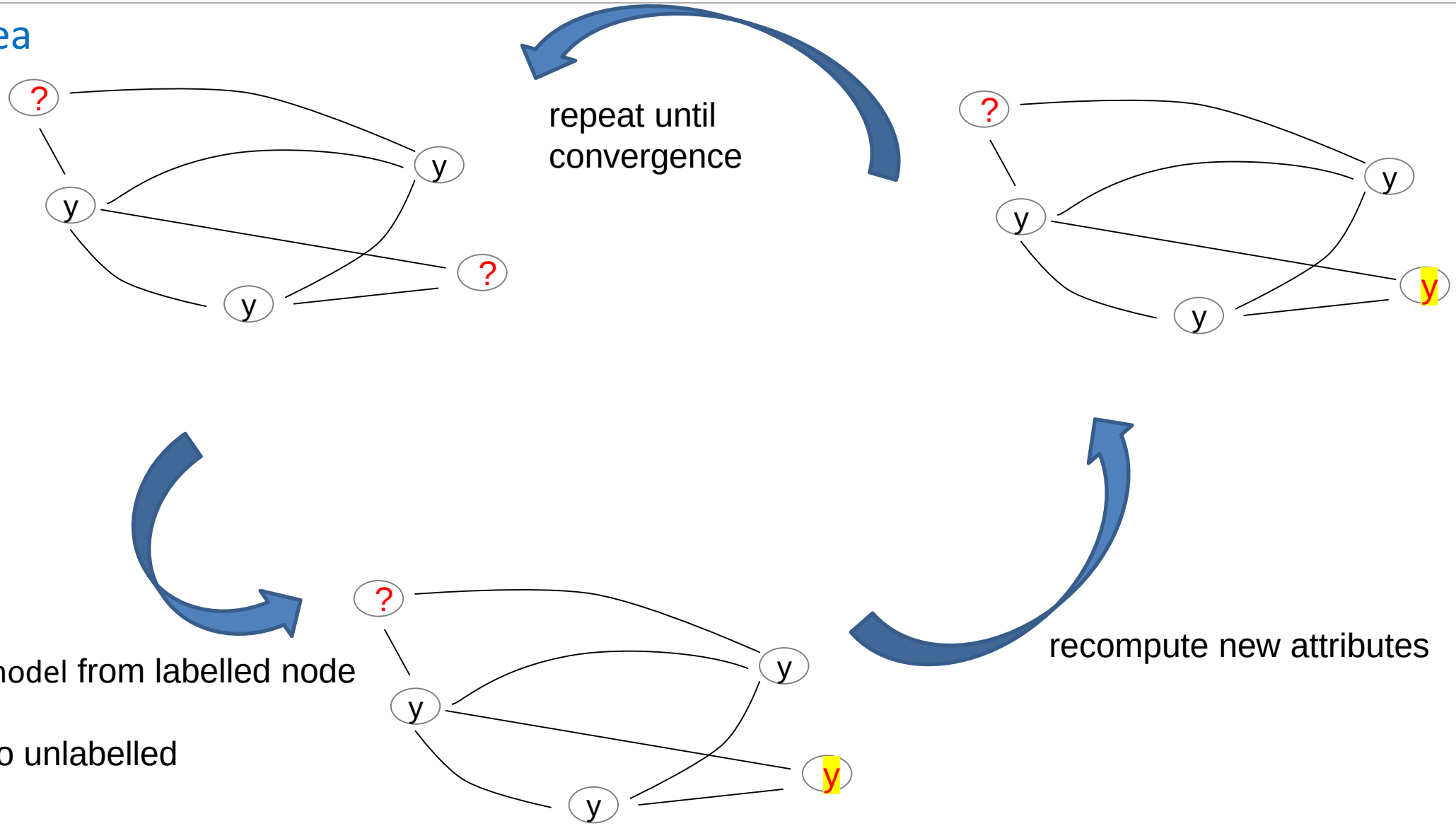
- Basic idea

  Network auto-correlation → Defining (new) node attributes which make a node aware about the distribution of the attributes X,Y on the linked nodes



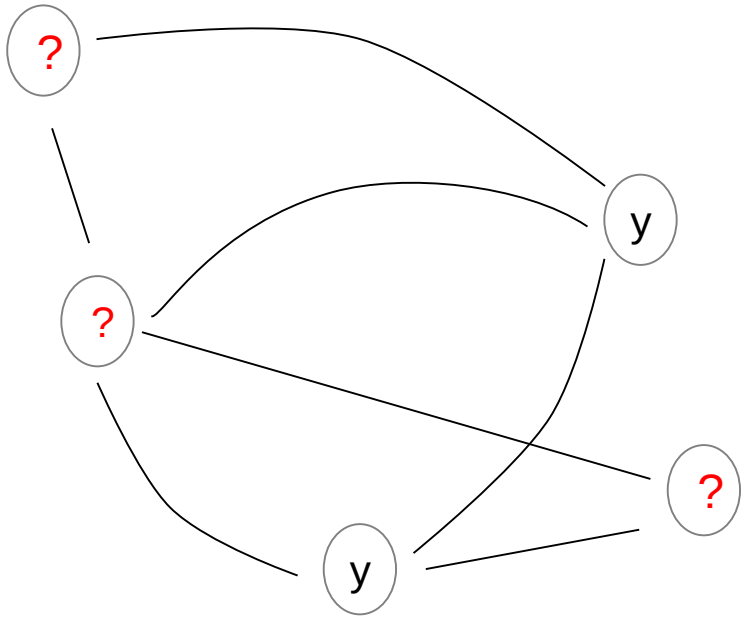
Iterative Convergence

- Basic idea



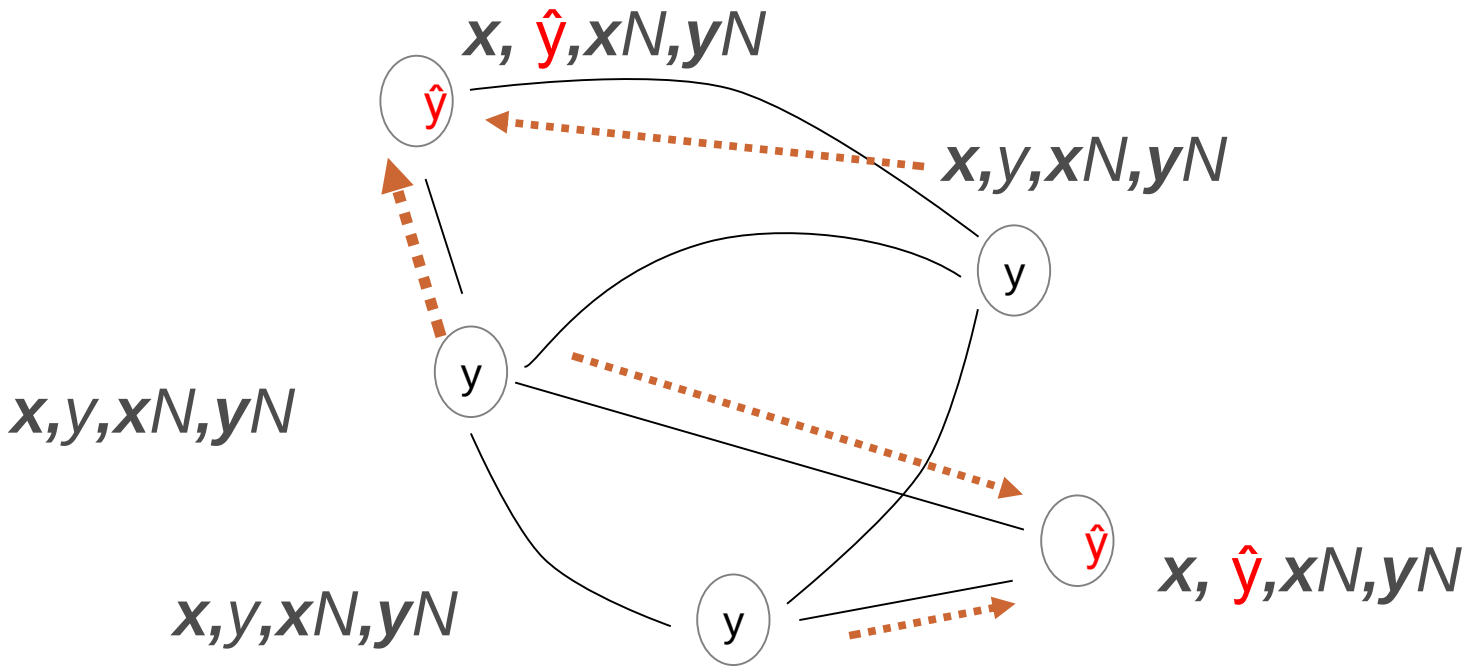
Issues

- Scarcely-labelled network
  - labeled nodes are possibly linked to unlabeled nodes and vice-versa.
  - both labeled and unlabeled nodes can be used to build a prediction of the unknown labels as more accurately as possible



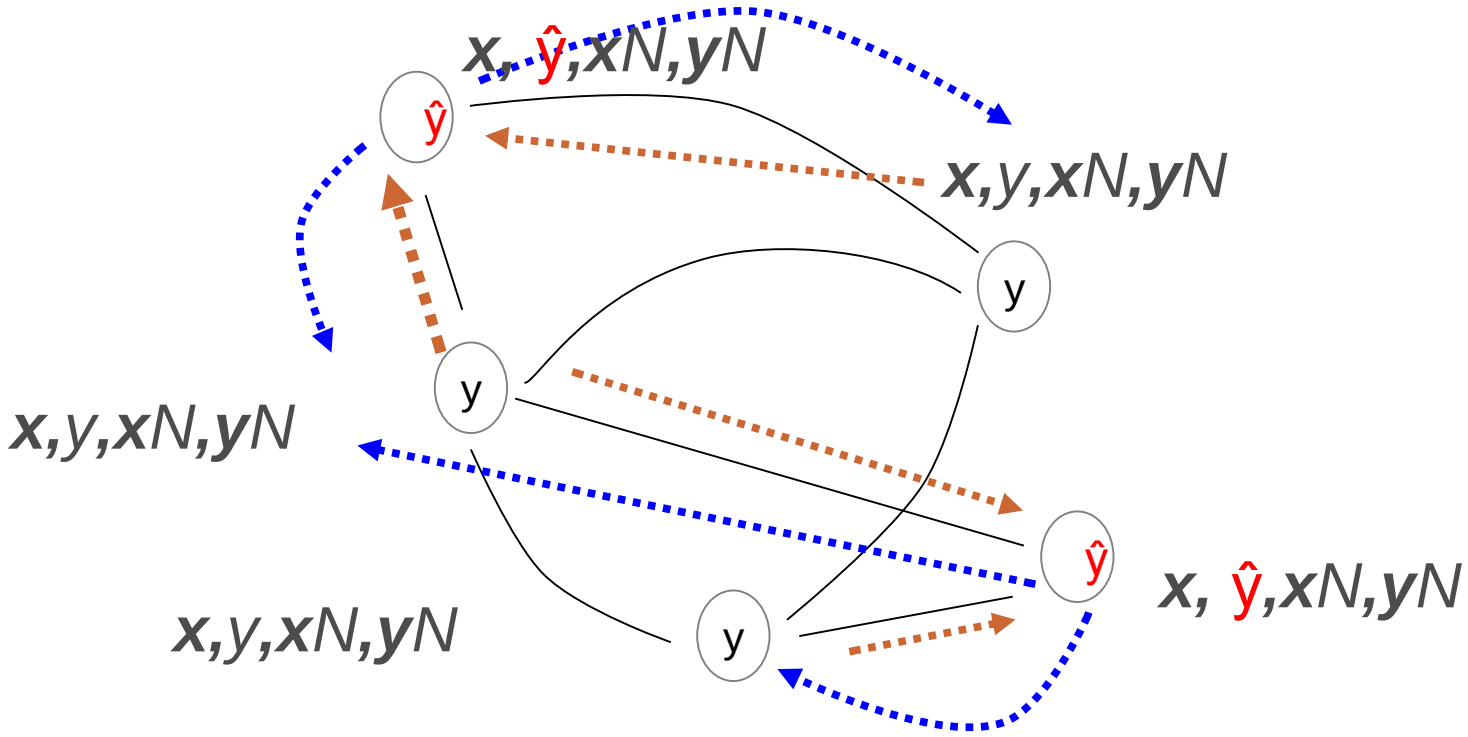
Iterative Convergence+
Collective Inference + Semi-supervised learning

- Basic idea
  - Joint predictions rather than single predictions.
  - Collective predictions on linked nodes can be used…



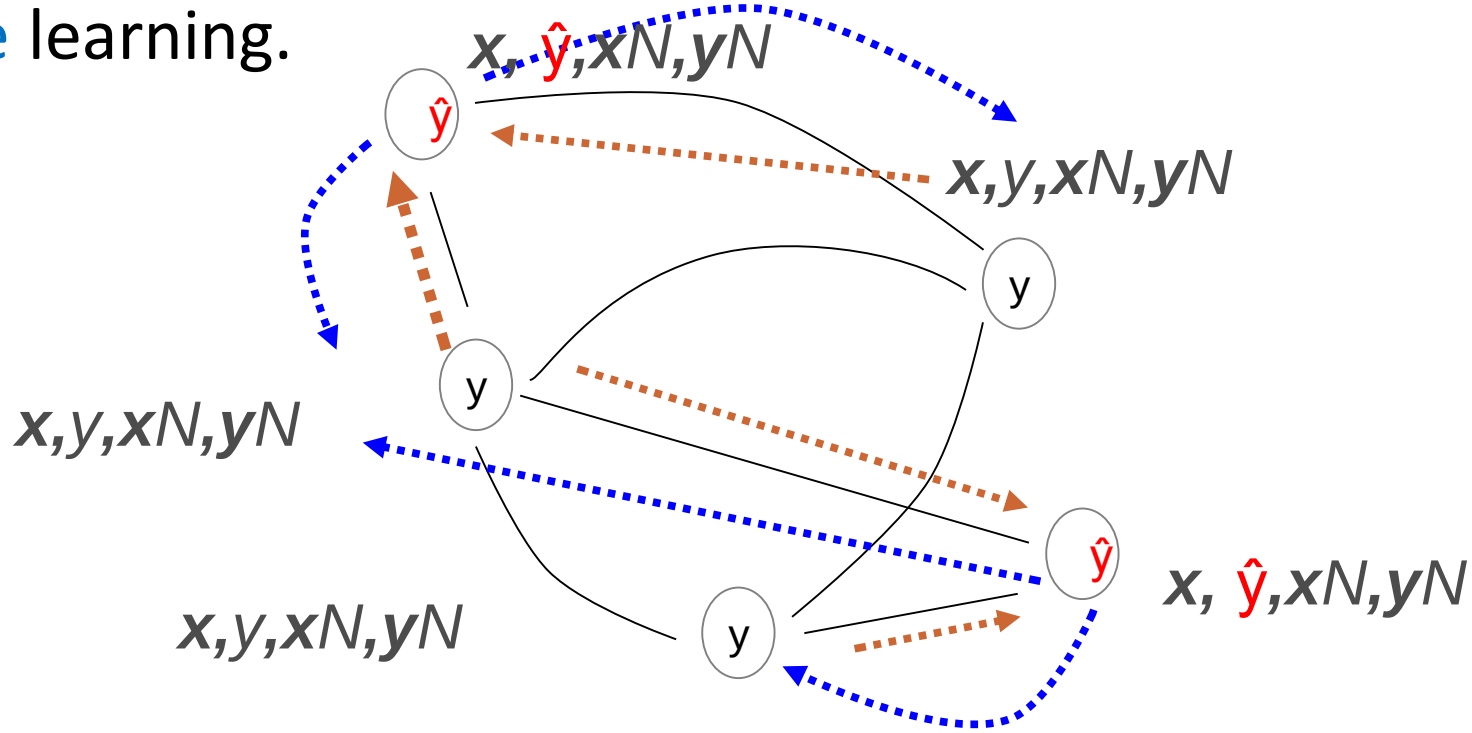
Iterative Convergence+
Collective Inference + Semi-supervised learning

- Basic idea
  - Joint predictions rather than single predictions.
  - …in order to mutually reinforce one label to each other.



Iterative Convergence+
Collective Inference + **Semi-supervised learning**

- Uses both labeled & unlabeled data to build classifier, whose goal is to classify only unlabeled data as accurately as possible.
- unlab. node not necessarily in the learning process, but used in other decisions of the iterative convergence
- no general rule valid for all possible instances is generated: semi-supervised is transductive learning.

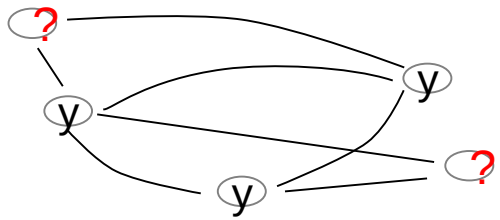


Iterative Convergence+
Collective Inference + Semi-supervised learning

- Procedure
  1. Re-build network structure
  2. Compute correlation-aware independent attributes XN.
  3. Initialize unknown labels with a base model learned on training data (X × Y).
  4. Determine correlation-aware dependent attributes YN.
  5. Learn a new model on training data (X × Y × XN × YN).
  6. Predict unknown labels and choose reliable labels.
  7. Updating YN, by accounting for new reliable labels.
  8. Iterate steps 4-7 until either the maximum number of iterations is performed or no new reliable label is estimated.

Correlation-aware attributes

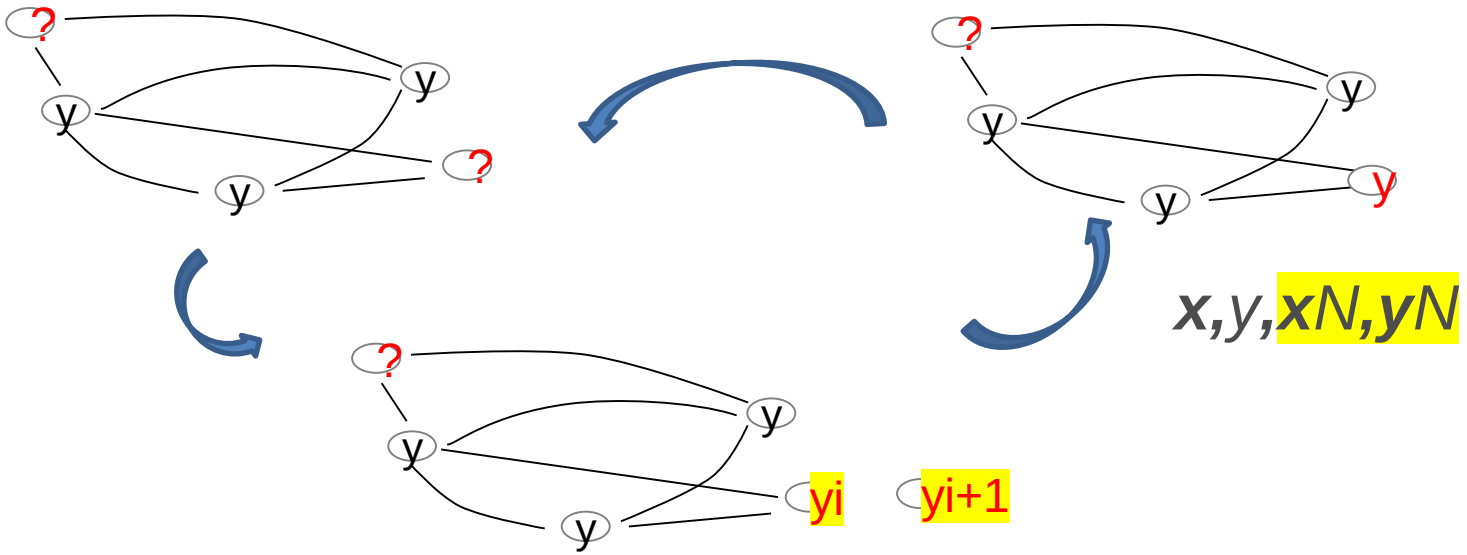
- discrete valued variables for classification
  - counts of labels, majority, mode,…
- numeric & discrete valued variables
  - weighted mean, standard deviation, histogram on the discretization-based ranges, ratio of weighted mean
- not necessary using all the linked nodes, but neighbour nodes
  - neighborhood defined with the adjacent nodes or with a maximum distance, computed with weighted edges, the neighbour should be within



x,y,xN,yN

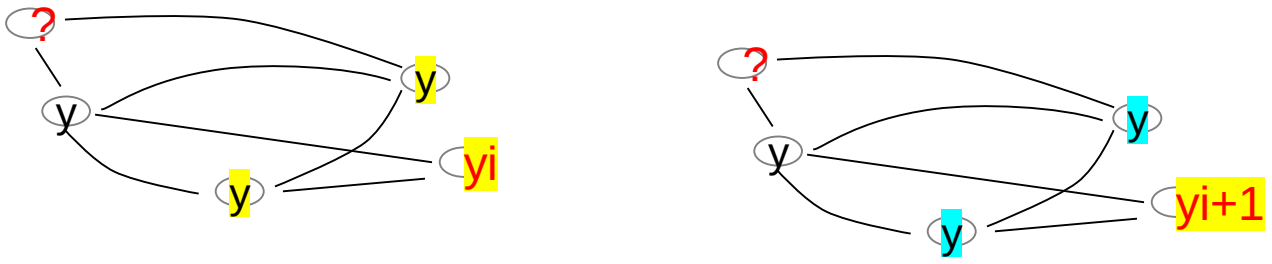
Label reliability

- Reliability of new predicted labels is estimated, in order to assign new labels to the nodes, and feed back the new labels into the learning process.
- Homophily principle: similar labels tend to be linked, so reliability can be quantified by a measure of the local auto-correlation of the label with respect to the training nodes.
- Two measures of local auto-correlation are used: local Moran Index, Getis-Ord index



Label reliability

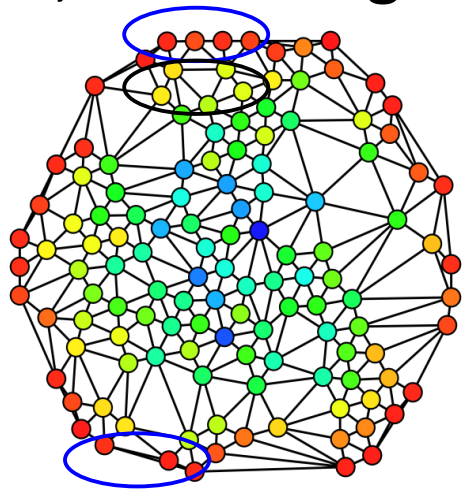
- For each node, let us consider the label predicted in the current iteration (yi+1) and the label previously assigned to the node (yi).
- Measure the local auto-correlation of the labels compared in the training network.
- Assign the label having the highest local auto-correlation to each node.



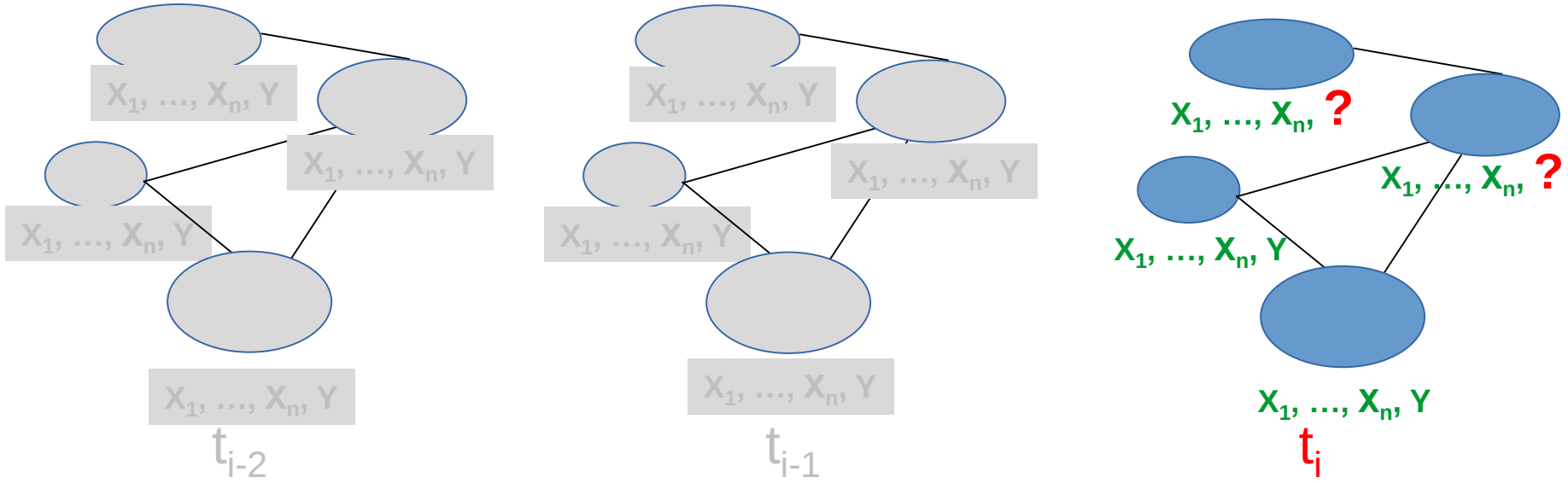
Re-build network structure

- in real-world networks, the auto-correlation is not the same for all the linked nodes:
  - pre-existing links may be unweighted, while we assume weight matrix
  - the auto-correlation is not uniform and depends on the edge weights

  →
  - build link structure based on attribute (dis)similarity
  - the higher the similarity among the attributes, the stronger the auto-correlation, the higher the weight

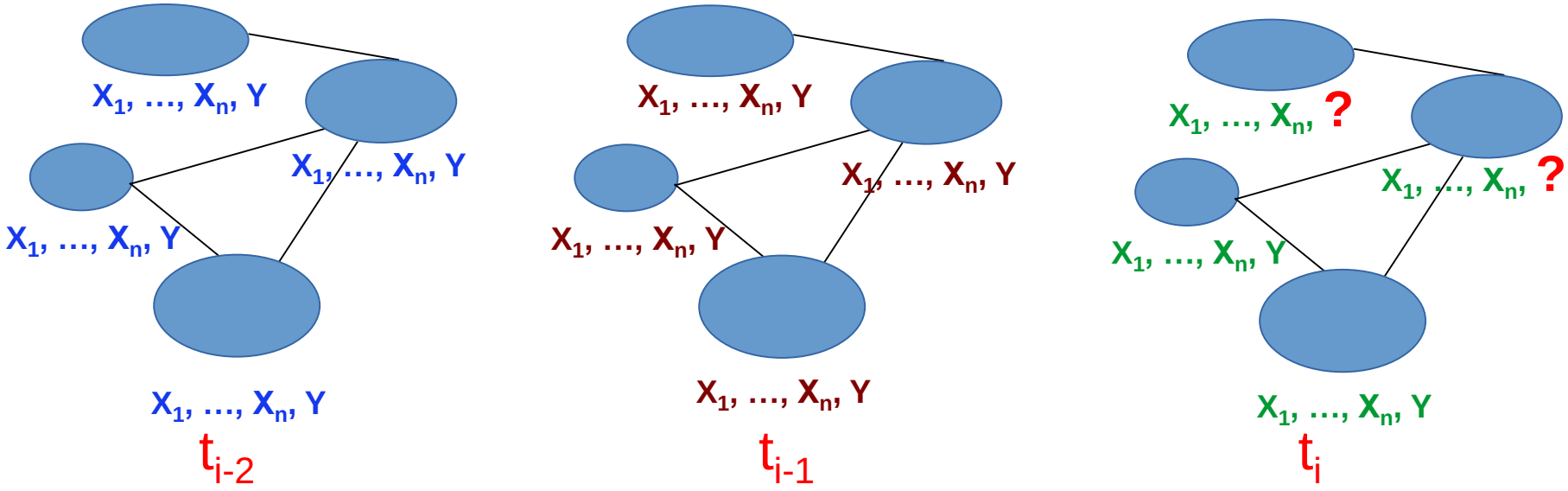


Node prediction in evolving networks



Node prediction in evolving networks

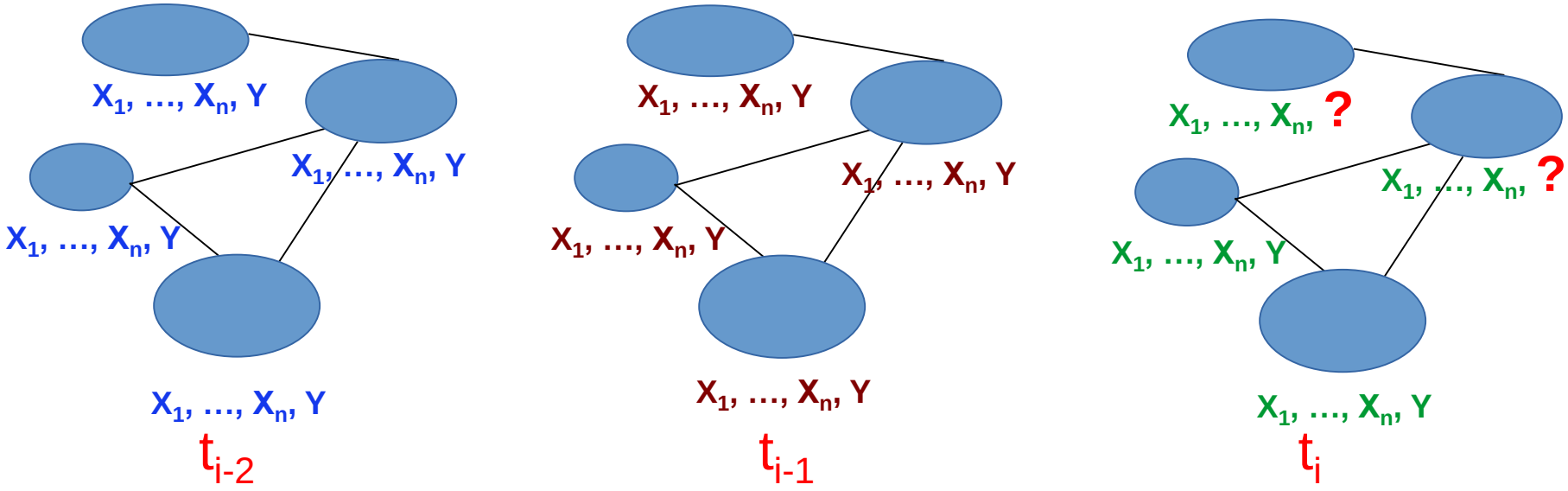
- Networks may change both in the structure and in the values of the attributes



- Network is partially labelled network at the time we observe, while it is fully labelled at previous times

Correlation in evolving networks

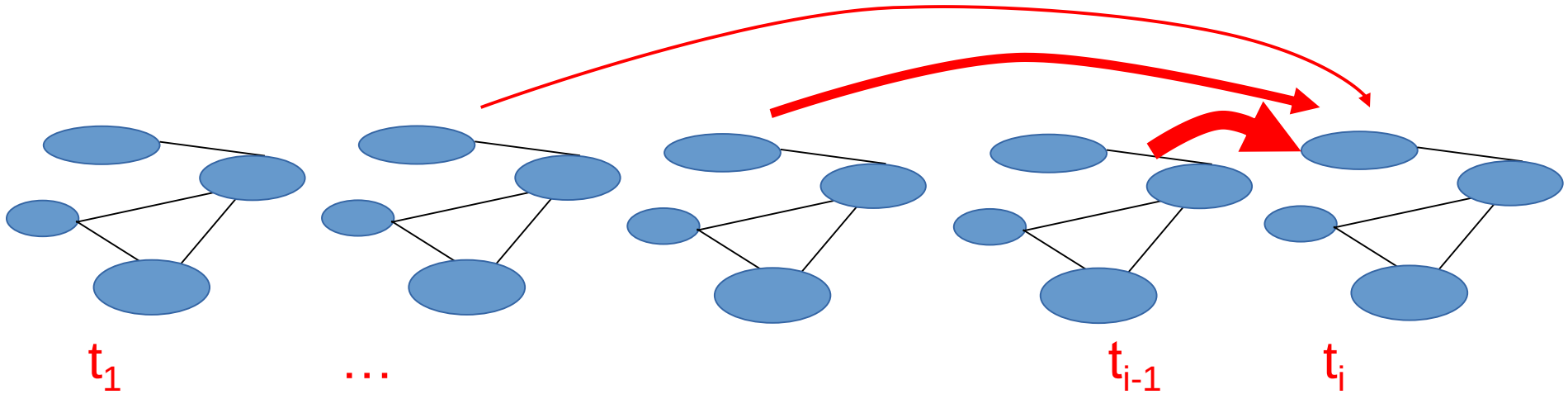
- Values of some attribute are correlated over a certain time lag



- Two sources of temporal correlation, temporal recurrence and temporal locality

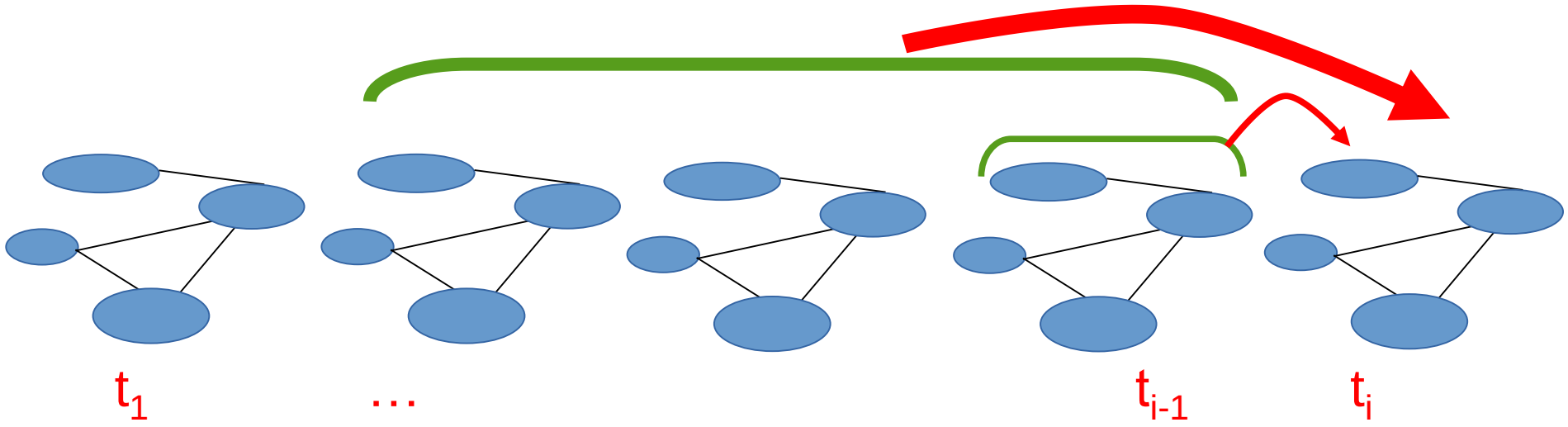
Correlation in evolving networks

- Values of some attribute are correlated over a certain time lag
- Temporal locality: values observed in the present are more highly correlated with the values of the recent past than those of the distant past



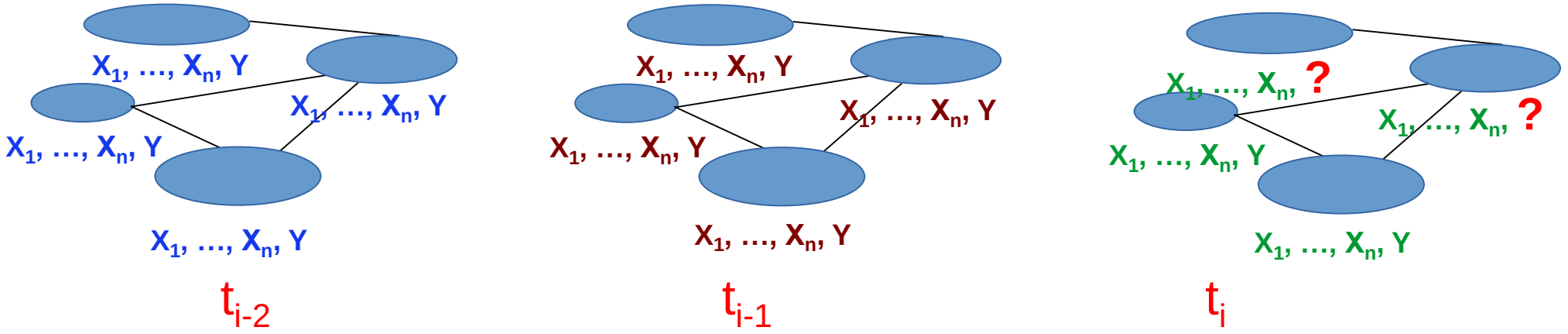
Correlation in evolving networks

- Temporal recurrence: Values are more influenced by a subsequence of values of the past than by a single value



Node prediction in evolving networks

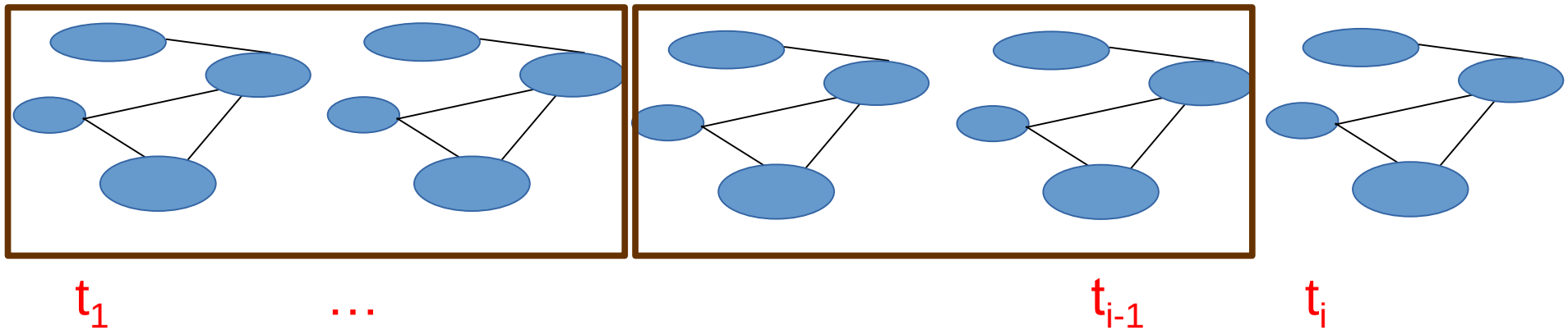
- Basic idea
- Two network learning scenario, fully labelled (t i-k,…,t i-1), partially labelled ti



- Learn prediction models specific for the network scenario
- Supervised for fully labelled, semi-supervised for partially lab.
- Ensemble of models, which accounts for the temporal correlation

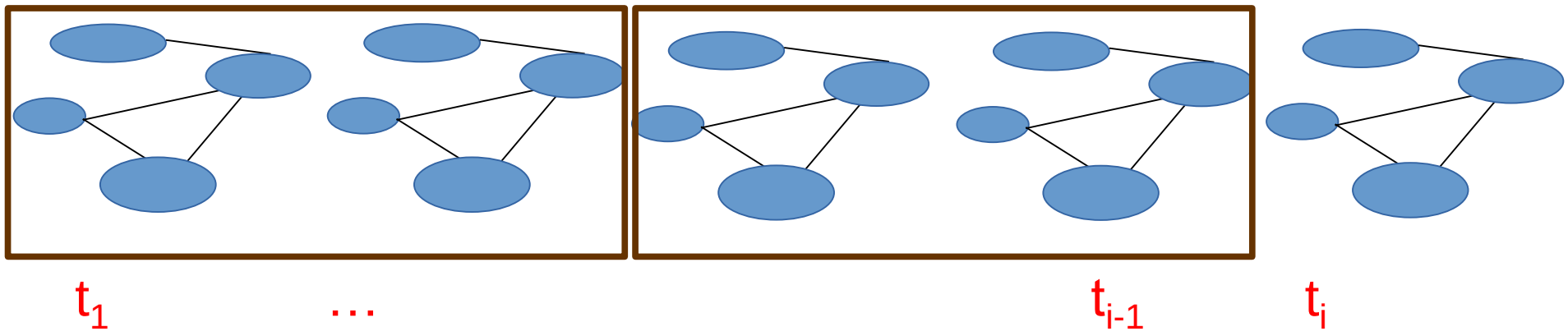
Node prediction in evolving networks

- Basic idea
- Temporal recurrence can be accommodated by learning models from subsequences of past networks, that is, time-windows of network data
- A window of the networks is synthesized in a summary network



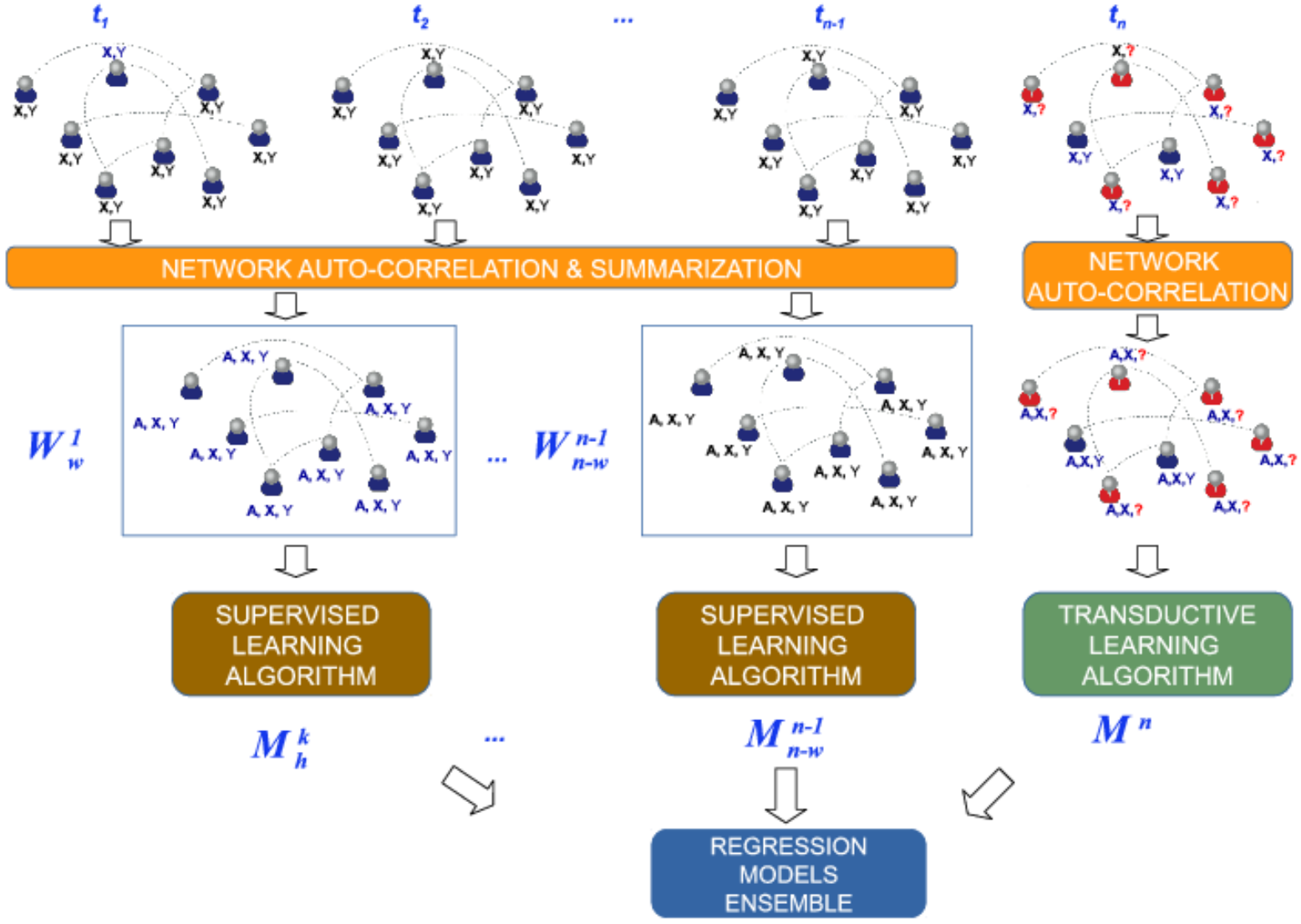
Node prediction in evolving networks

- Basic idea
- Temporal locality can be accommodated with weighting schemes based on the proximity temporal w.r.t. the current network
- Uses weighting schemes to build the ensemble and summary networks



Node prediction in evolving networks

- Procedure



Node prediction in evolving networks

- Procedure
  - Compute correlation-aware attributes XN and YN, in order to account for auto-correlation within networks
  - Generate summary networks from historical network data
  - Learn regression models on summary networks (supervised)
  - Learn regression model on current network (semi-supervised)
  - Build an ensemble E by using the prediction models:
    - higher weights to the models closer to the current network, and lower weights to those more distant.

$$E(x) = \sum_j \left(\frac{1}{t_m - t_j + 1} \right) \lambda f_j(x)$$

References

- Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, Tina Eliassi-Rad: Collective Classification in Network Data. AI Magazine 29(3): 93-106 (2008)

- Corrado Loglisci, Annalisa Appice, Donato Malerba: Collective regression for handling autocorrelation of network data in a transductive setting. J. Intell. Inf. Syst. 46(3): 447-472 (2016)

- Corrado Loglisci, Donato Malerba: Leveraging temporal autocorrelation of historical data for improving accuracy in network regression. Statistical Analysis and Data Mining 10(1): 40-53 (2017)