A Scalable Multiclass Algorithm for Node Classification

Giovanni Zappella, Università degli Studi di Milano
giovanni.zappella@unimi.it

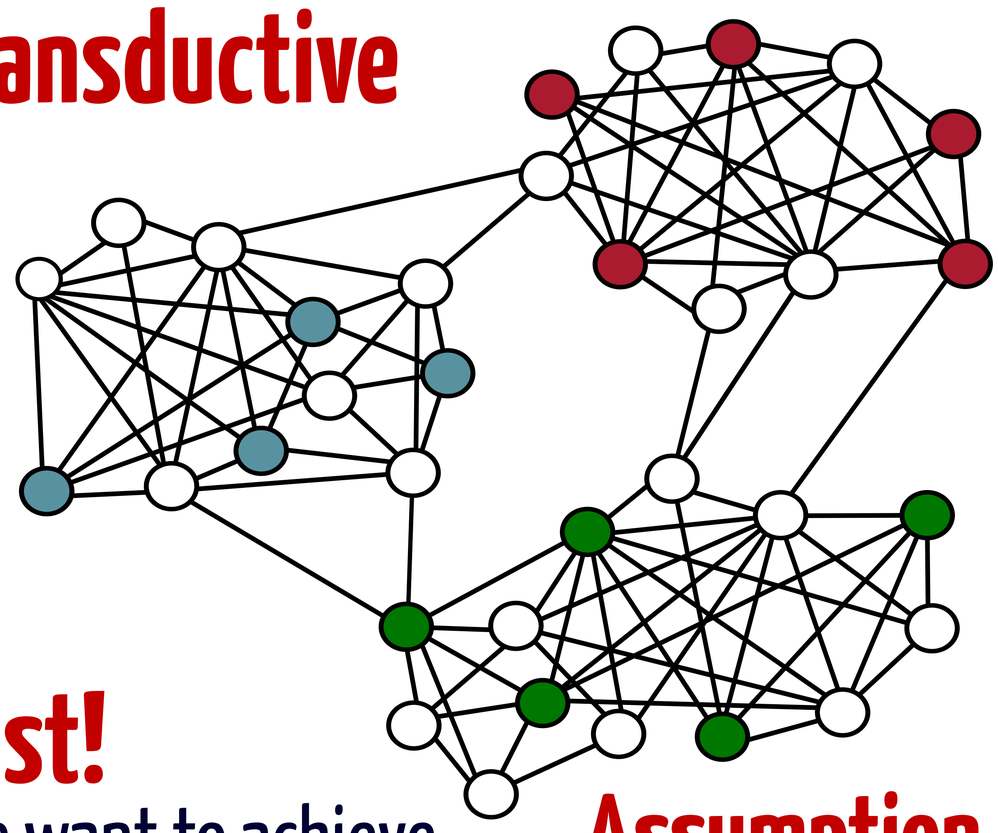
The problem

Given a weighted graph and a partial node labeling, we want to predict the missing node labels.

This can be useful for:
- social networks (people and relations)
- WWW (pages and similarity)
- biological networks

Most of these networks are huge and for practical applications scalability is a 'condicio sine qua non', not just an important feature.
Like many other node classification algorithms we exploit a widespread property called 'homophily'.
In practice we expect that well-connected nodes are similar (often have the same label).

Transductive



Fast!
We want to achieve both **speed and accuracy**

Assumption
Well-connected nodes are similar

The Game

The Graph Transduction Game (GTG) was introduced by Erdem and Pelillo in 2011 [1].

Each node is a player in a non-cooperative polymatrix game. Each pair of connected nodes has a diagonal payoff matrix where the nonzero payoffs are exactly the weights of the edges connecting the nodes.
Each player can choose its strategy independently and each label is a pure strategy for the player.
In practice, given the payoff matrix described above, the players get a non-zero reward only if both of them choose the same strategy (label).

Erdem and Pelillo found the Nash equilibrium of the game using Evolutionary Stable Strategies.
Their algorithm (we call it GTG-ESS) seems to have a time complexity **cubic** in the number of nodes

MUCCA algorithm

MUCCA is a fast algorithm that finds a Nash Equilibrium of the GTG game on a tree.

The learning protocol is batch: we have train/test split so some nodes will not be able to change their strategy (label) in any case. The remaining nodes will adopt the label that is more convenient for them.

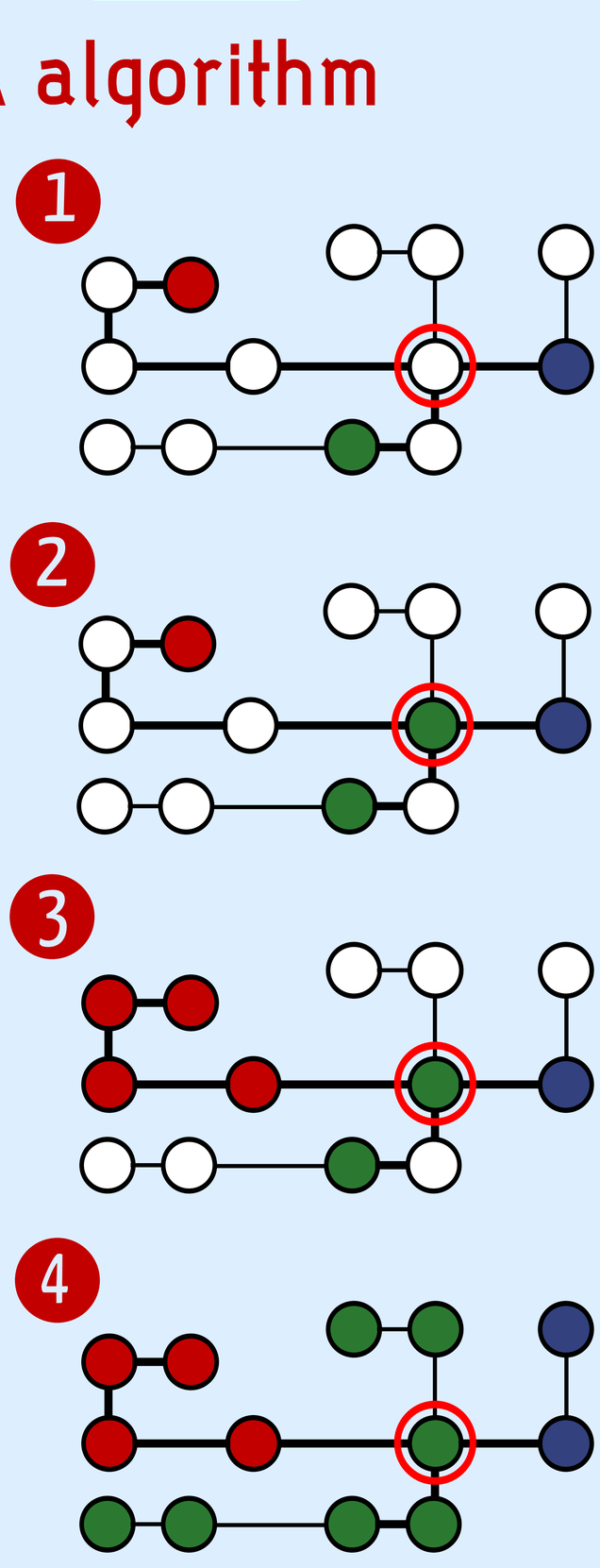
In the **first phase**, MUCCA finds a path among all the labeled nodes (marked with strong black line).
After this operation all the nodes with at least three black-line edges incident to them are called "forks".

Once the forks have been located, the **second** phase starts.
We estimate the label of each fork node using mincut.

In the **third** phase we label all the nodes on the paths connecting labeled nodes using mincut.

In the **fourth** phase, all the nodes that are not labeled are found in subtrees connected to a node on the path between two labeled nodes (root of the subtree). All the nodes in the subtrees get the same label of the root of their tree.

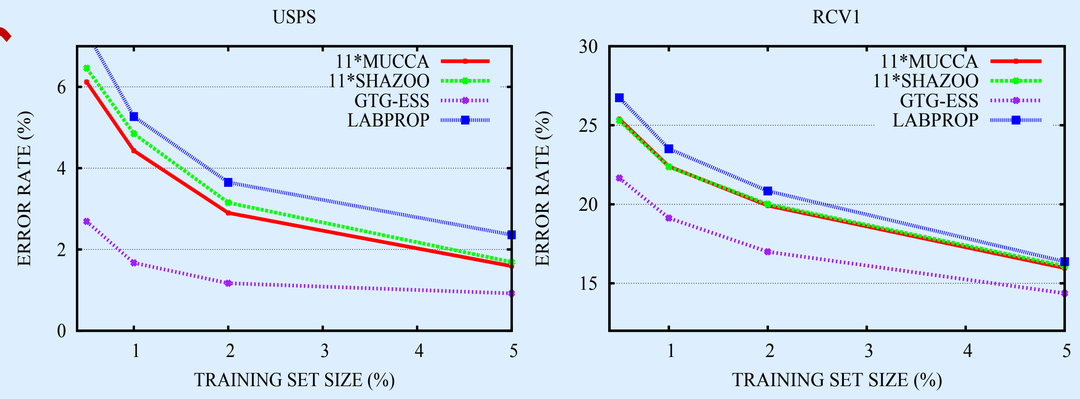
This simple algorithm finds a Nash Equilibrium of GTG on a tree.



Experiments

More experimental results are available in the paper

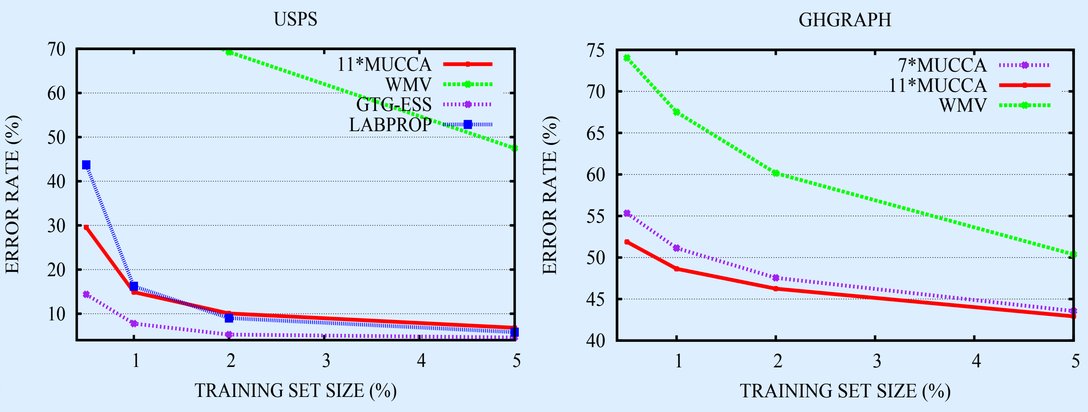
Binary



USPS and RCV1 are quite well-known datasets. All the graphs were generated using k-NN (k=10) and Gaussian weights.
GHGRAPH is a graph of GitHub repositories, whose edges are determined by co-commits of programmers and whose labels are determined by the main programming language used in the repository.

n*MUCCA is a committee of n MUCCA instances, the final prediction is a majority vote over the committee.

Multiclass



Results are averaged on 10 runs

Short Bibliography

[1] Erdem and Pelillo -- Graph Transduction as a Non-cooperative Game, Workshop on Graph-based Representations in Pattern Recognition 2011
[2] Cesa-Bianchi, Gentile, Vitale and Zappella -- See the tree through the lines: the Shazoo algorithm , NIPS 2011
[3] Cesa-Bianchi, Gentile, Vitale and Zappella -- Random spanning trees and the prediction of weighted graphs , ICML 2010
[4] Zhu and Ghahramani -- Semi-supervised learning using gaussian fields and harmonic functions, ICML 2003