

Active Learning for Hidden Attributes in Networks

Xiao Ran YAN, Yao Jia ZHU, Jean-Baptiste ROUQUIER, Christopher MOORE

In many networks, vertices have hidden attributes (or *types*) that are correlated with the network's topology. For instance, in social networks, people are more likely to be friends if they are demographically similar (this is called an *assortative* network). In food webs, predators typically eat prey of lower body mass. We explore a setting in which the network's topology is known, but these types are not. (The reciprocal setting, in different studies, would be to know the types and predict the edges.)

Active learning means that each vertex can be queried, learning the value of its hidden type — but only at some cost, say, by devoting resources in the laboratory or the field. We thus need an algorithm to choose which vertices to query, in order to learn as much as possible about the types of the remaining vertices. We do this one vertex at a time.

Block model

We assume that the network is generated by the following probabilistic model, but our method can be adapted to a wide range of probabilistic models in which topology is correlated with hidden types of the vertices.

- Each vertex v has a hidden type $t(v) \in \{1, \dots, k\}$. Let n_i be the number of nodes of type i .
- Between each pair of vertices u, v , there is an edge with probability $p_{t(u), t(v)}$; and these events are independent. Let e_{ij} be the number of edges from type i to type j .

Given an labeling t and probabilities p_{ij} , the likelihood of generating G is

$$\mathcal{L}(G|t, p) = \prod_{i,j=1}^k p_{ij}^{e_{ij}} (1 - p_{ij})^{n_i n_j - e_{ij}}$$

We assume a uniform prior on the probabilities p_{ij} : they are chosen independently and uniformly from $[0; 1]$. The likelihood of a labeling is thus:

$$\mathcal{L}(G|t) = \int \int_{i,j=1}^k \int_{p_{ij}=0}^1 \mathcal{L}(G|t, p) dp_{ij}$$

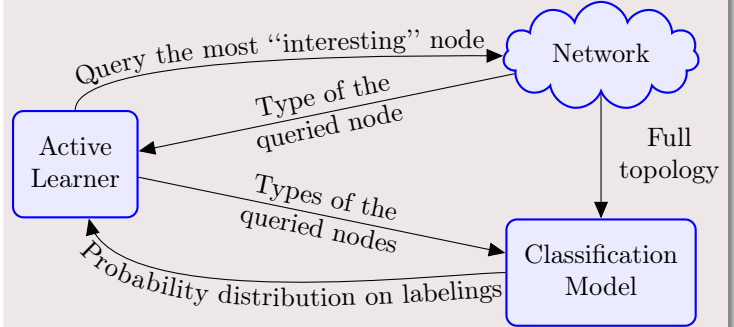
It is highest when e_{ij} is close to 0 or to $n_i n_j$, its maximum.

We use a Bayesian block model and look for a labeling that maximizes the likelihood of the given graph.

Assortative and disassortative networks An example of a *disassortative* network is the graph with airports as nodes and flights as links: small airports tend to be connected to hubs. Large p_{ii} means assortative, while large p_{ij} for $i \neq j$ means disassortative. Our method works for both.

Algorithm

The typical usage of the algorithm is to let it query a fixed number of vertices, then stop it. The output of the algorithm is then a probability distribution d on the labelings.



Two methods to choose the most interesting node

Mutual information $MI(v)$

A classical approach in active learning is to query the vertex v with the largest **mutual information** between its type and that of the others. Which means (after a short derivation) a vertex about which we are quite uncertain, but which is strongly correlated with other vertices.

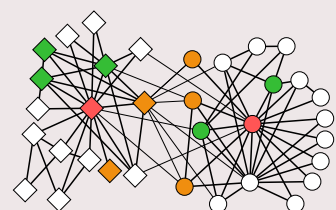
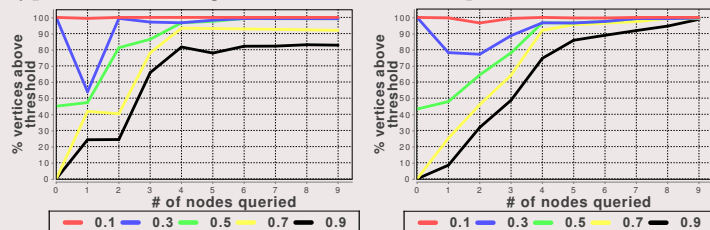
Average Agreement $AA(v)$

The *overlap* between two labelings is the number of vertices on which they agree. For a vertex v , draw two labelings according to d , conditioned on the fact that they agree on v , and define $AA(v)$ as their expected overlap.

We estimate AA and MI by sampling from the space of labelings according to the probability distribution d .

Results

We choose two networks with known types (the classical Zachary's Karate club and a food web of 488 species), and hide the types from the algorithm to see how it performs.



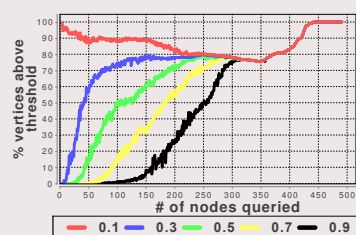
Zachary's Karate club. Colors: Nodes consistently queried first, nodes often queried afterwards, nodes usually queried last.

Both heuristics appear to explore the network intelligently, first querying vertices at the centers of communities, then vertices along the boundaries between communities, and last vertices deep inside their community, with no doubt about their type.

Karate club social network. Left: MI, right: AA.

The y axis shows the fraction of vertices, other than those queried so far, which are labeled correctly by the distribution d , with various probabilities (0.1, 0.3, 0.5, 0.7, 0.9).

AA is almost perfect after 9 queries. AA is slightly better than MI, and both are better than simple heuristics.



Foodweb, with AA heuristic.

After querying about half the species, the curves collapse: each vertex is predicted correctly with probability either greater than 0.9 or less than 0.1. In other words, the algorithm is almost certain about all the vertices, but wrong about many of them. Most of these are species which are poorly modeled by the block model: they would be misclassified even if you knew the types of all other species.