Active Learning for Hidden Attributes in Networks

Xiao Ran Yan, Yao Jia Zhu,
Jean-Baptiste Rouquier, Cristopher Moore

ISC PIF.fr

Plan

Nodes with attributes

Examples :

- Social network : demographics (gender, town, hobbies...)
- Food web : body mass, habitat...

They are :

- correlated with topology
- costly to determine : go to the field, lab experiment, phone poll...

The problem

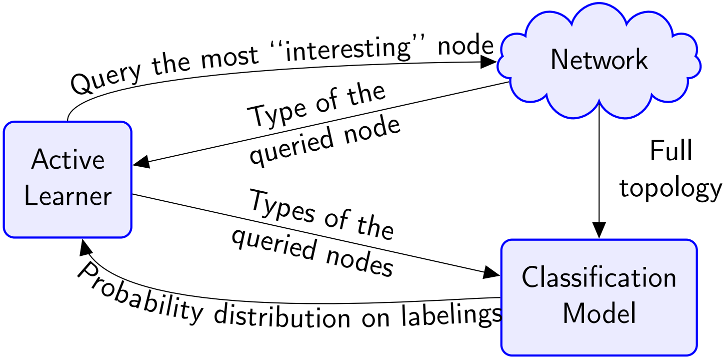
Classic problem : partial data on a network, guess the missing part.
Two settings :

- Known node attributes, unknown links.
  Request : "Is A connected to B ?".
- Known links, unknown node attributes.
  Request : "What is the type of A ?"
  This is what we focus on.

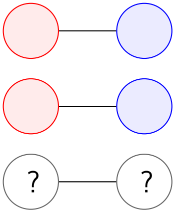
Limited number of requests.
Active learning : 1/ query, 2/ update the representation, 3/ go back to 1.
Need to carefully choose the next request.

Plan

Let the algorithm request a fixed number of vertices, then stop it.
The output is then a probability distribution on the labelings.

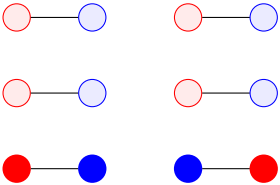


Probability distribution on labelings

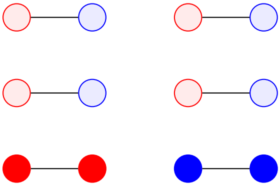


$\Longrightarrow$

- high probability :

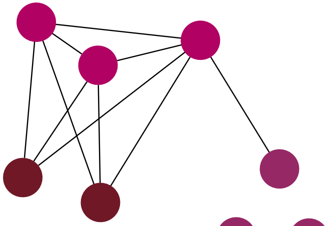


- low probability :



Plan

A network generation model



The node types determine the edges :

- Fixed set of nodes.
- Each vertex v has a hidden type $t(v) \in \{1, ..., k\}$.
  n_i : number of type i nodes.
- Fixed set of probabilities $(p_{ij})_{1 \leqslant i,j \leqslant k}$.
- Independently draw each edge u, v with probability $p_{t(u),t(v)}$.
  e_{ij} : number of edges from type i to type j.

Assortative and disassortative networks

This model works for both.

- biological networks tend to be disassortative : large degree nodes have links to small degree ones.
- social networks tend to be assortative : members connect with people demographically similar

Large p_{ii} means assortative, while large p_{ij} for $i \neq j$ means disassortative.

Bayesian

- Given a labeling t and probabilities $p = (p_{ij})_{1 \leqslant i,j \leqslant k}$, the likelihood of generating the graph G is

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

Bayesian

- Given a labeling t and probabilities $p = (p_{ij})_{1 \leqslant i,j \leqslant k}$, the likelihood of generating the graph G is

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

- Guess t and p at the same time, with a Bayesian model : look for (t, p) that maximizes the likelihood of G.
- We are not interested in p, so assume a uniform prior : p_{ij} i.i.d. in $[0; 1]$.

Bayesian

- Given a labeling t and probabilities $p = (p_{ij})_{1 \leqslant i,j \leqslant k}$, the likelihood of generating the graph G is

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

- Guess t and p at the same time, with a Bayesian model : look for (t, p) that maximizes the likelihood of G.
- We are not interested in p, so assume a uniform prior : p_{ij} i.i.d. in $[0; 1]$. The likelihood of a labeling becomes :

$$\mathcal{L}(G|t) = \iint_{i,j=1}^{k} \int_{p_{ij}=0}^{1} \mathcal{L}(G|t,p) \mathrm{d}p_{ij} = \ldots = \prod_{i,j=1}^{k} \frac{1}{(n_i n_j + 1)\binom{n_i n_j}{e_{ij}}}$$

  It is highest when e_{ij} is close to 0 or to $n_i n_j$, its maximum.
- Distribution on labelings $\mathbb{P}(t) \propto \mathcal{L}(G|t)$.
  Markov chain Monte Carlo to estimate it.

Lost ?

Block model isn't the main part, our method can be adapted to other probabilistic models in which topology is correlated with hidden types.

Plan

Plan

Classical approach in active learning : request the vertex v with the largest mutual information between its type and that of the others.

$$MI(v) := I(v; G \setminus v) = H(v) - H(v \mid G \setminus v)$$

- large $H(v)$: we are uncertain about v
- small $H(v \mid G \setminus v)$: v is strongly correlated with other vertices

Plan

Motivation

Definitions :

- *overlap* $|t_1 \cap t_2|$: number of vertices on which two labelings t_1 and t_2 agree.
- d : probability distribution on labelings (according to model and known labels).

Motivation

Definitions :

- *overlap* $|t_1 \cap t_2|$: number of vertices on which two labelings t_1 and t_2 agree.
- d : probability distribution on labelings (according to model and known labels).

Ideally, maximize $|t_1 \cap t_2|$ where $\begin{cases} t_1 \text{ drawn from } d \\ t_2 \text{ the real labeling} \end{cases}$.

t_2 unknown, approximate with d as well.
Thus, choose v maximizing $|t_1 \cap t_2|$ once $t(v)$ is known.

In short

For a vertex v, draw two labelings according to d, conditioned on the fact that they agree on v, and define the average agreement $AA(v)$ as their expected overlap.

In short

For a vertex v, draw two labelings according to d, conditioned on the fact that they agree on v, and define the average agreement $AA(v)$ as their expected overlap.

$$AA(v) = \frac{\sum_{t_1,t_2:\ t_1(v)=t_2(v)} P(t_1)P(t_2)\,|t_1 \cap t_2|}{\sum_{t_1,t_2:\ t_1(v)=t_2(v)} P(t_1)P(t_2)}$$

Plan

Performance indicator

Two data sets with known types. We hide the types from the algorithm to test it.

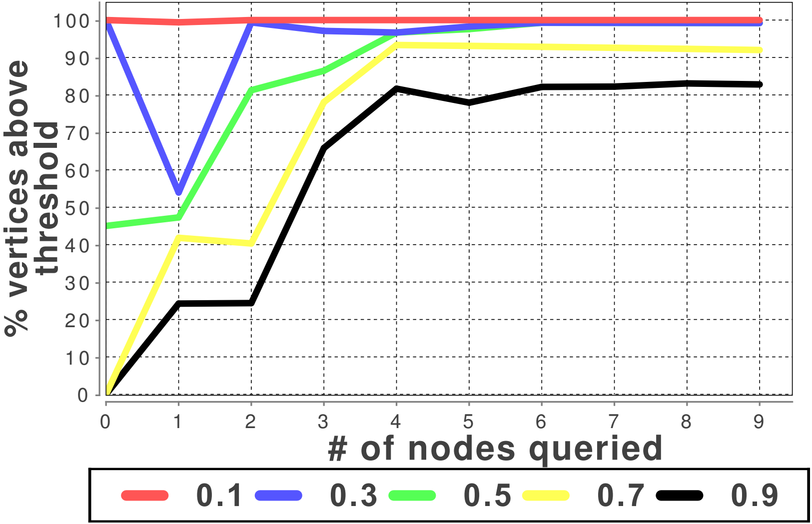
After r requests, for a given vertex, estimate with what probability the Gibbs distribution assigns it the correct type.

For a threshold, q, estimate the proportion of vertices being assigned the correct type with probability at least q.

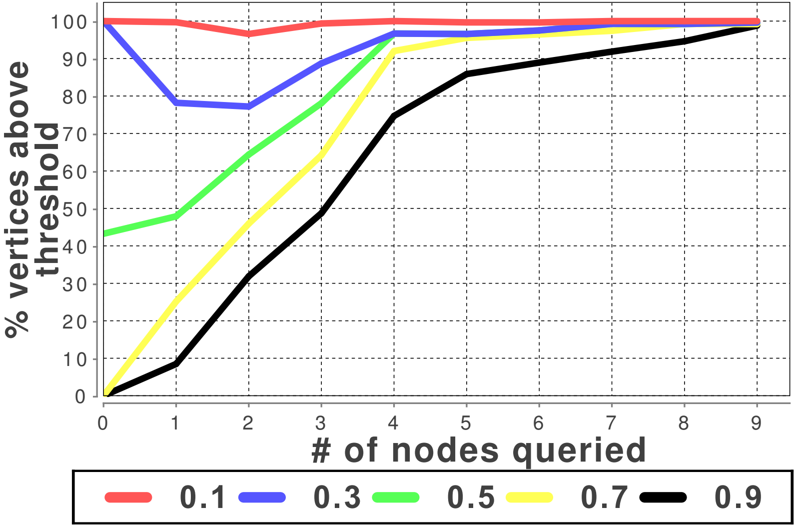
Plot this as a function of r.

MI and AA better than simple heuristics.

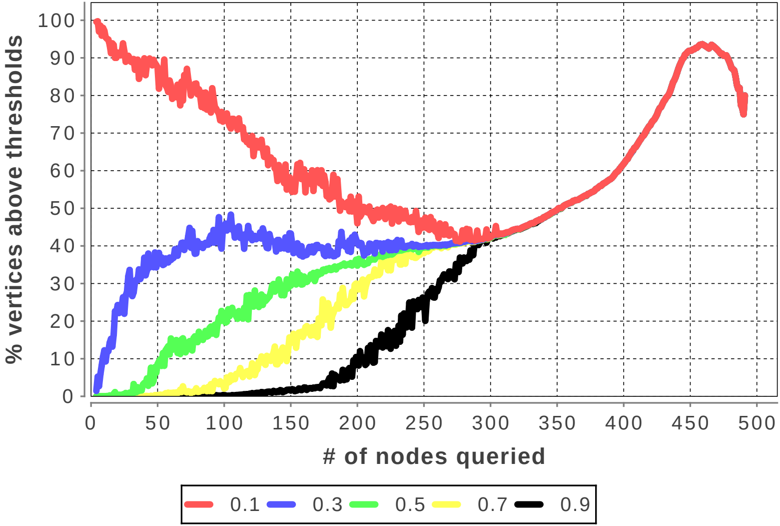
Zachary's Karate club, MI



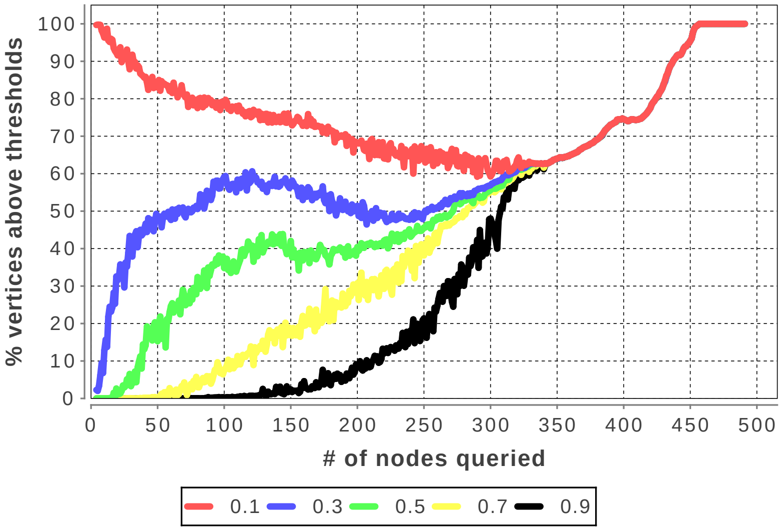
Zachary's Karate club, AA



Foodweb, MI


% vertices above thresholds
100
90
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
300
350
400
450
500
# of nodes queried
0.1
0.3
0.5
0.7
0.9

Foodweb, AA

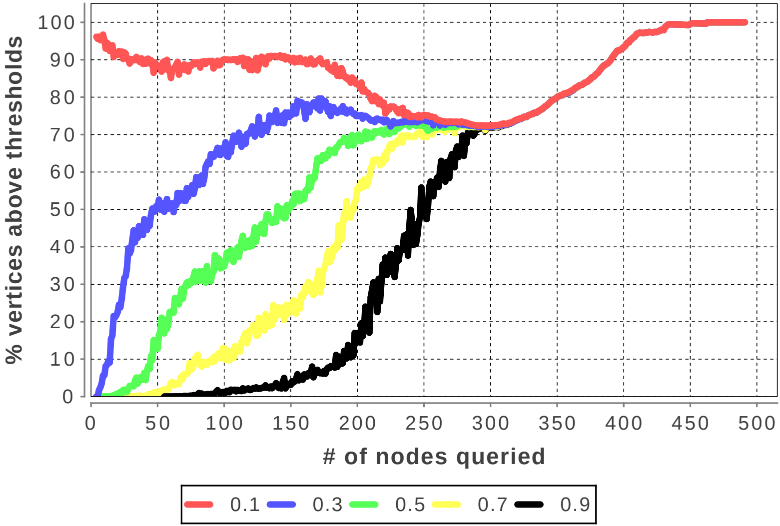

% vertices above thresholds
# of nodes queried
0 10 20 30 40 50 60 70 80 90 100
0 50 100 150 200 250 300 350 400 450 500
0.1
0.3
0.5
0.7
0.9

Collapse of the curves

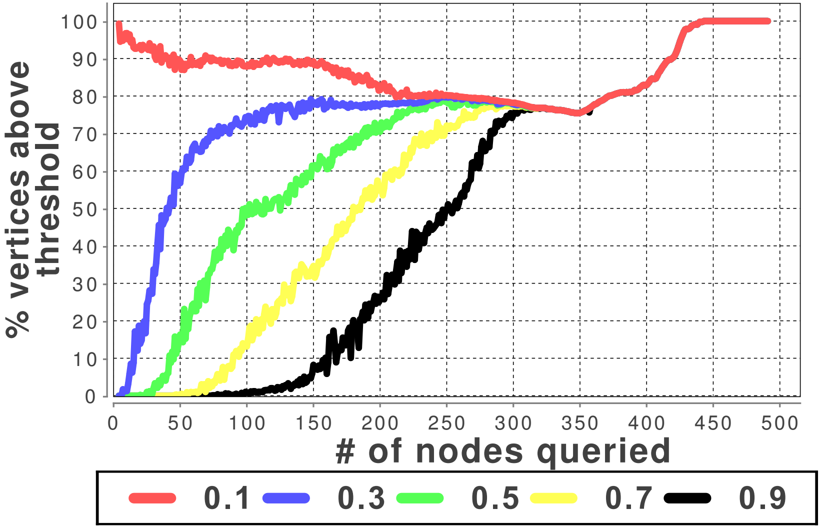
Superimposed curves means correct prediction with probability $\geqslant 0.9$ or $\leqslant 0.1$. Either right most of the time, or wrong most of the time, about each vertex : almost certain about all the vertices, but wrong about many of them.

Most of these "unknown unknowns" are species poorly modeled by the block model using habitat as the only type. They would be misclassified even if you knew the types of all the other species.

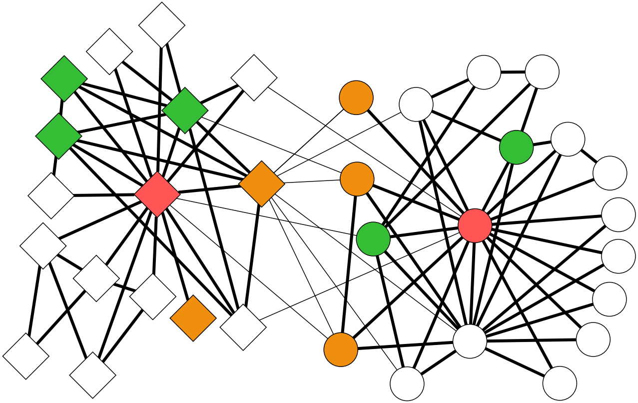
Foodweb, MI


% vertices above thresholds
100
90
80
70
60
50
40
30
20
10
0
0 50 100 150 200 250 300 350 400 450 500
# of nodes queried
0.1 0.3 0.5 0.7 0.9

Foodweb, AA


% vertices above threshold
100
90
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
300
350
400
450
500
# of nodes queried
0.1
0.3
0.5
0.7
0.9

Query order (Zachary's Karate club)



Nodes consistently queried first (community center),
nodes often queried afterwards (boundary),
nodes usually queried last (obvious type).

Future work

- other data sets
- heterogeneous degree distribution
- other network generation models