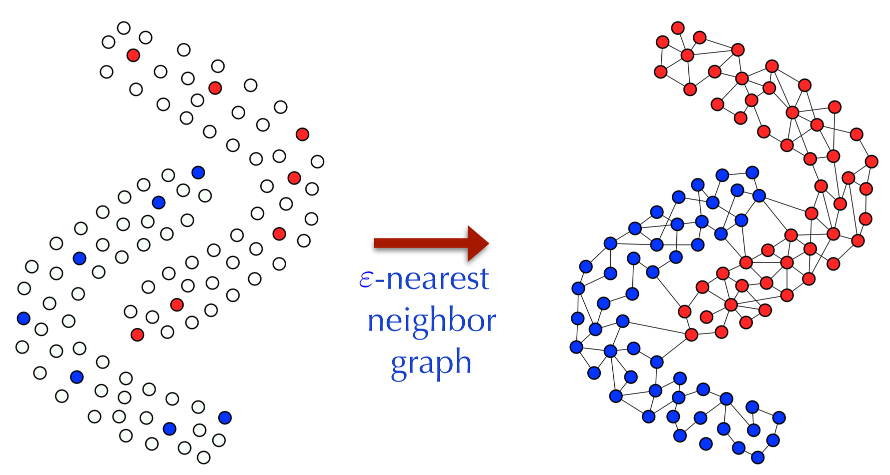
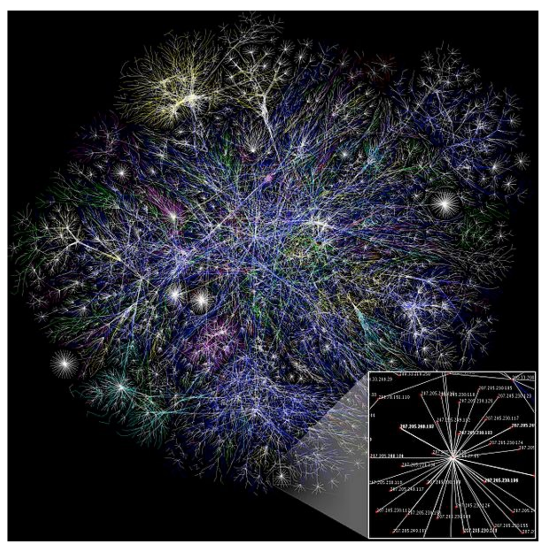
S^2 : An Efficient Graph Based Active Learning Algorithm with Application to Nonparametric Classification

Gautam Dasarathy gautamd@cs.cmu.edu
Robert Nowak rdnowak@wisc.edu
Xiaojin Zhu jerryzhu@cs.wisc.edu
Carnegie Mellon University

Label Prediction on Graphs

Important problem that underlies many machine learning tasks





Semi-Supervised Learning problems can be converted to graph label prediction problems

(Social) Network Analysis: Label vertices on a graph (spam/not spam, republican/democrat) by querying a few vertices

Problem Setup

$G = ([n], E)$ is a known graph on vertex set $[n] = \{1, 2, \ldots, n\}$.

$f : [n] \to \{-1, +1\}$: Unknown Labeling function (oracle).

Goal : Sequentially and actively select a subset $L \subset [n]$

Observe $f(L) = \{f(i) : i \in L\}$

Predict $f(L^c) = \{f(i) : i \notin L\}$

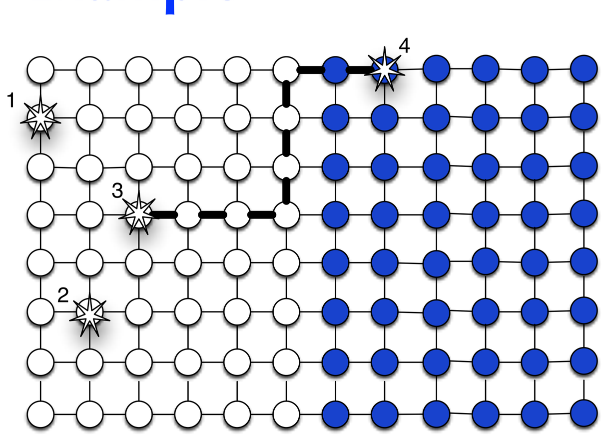
Questions

- How big does L have to be?
- How can L be chosen efficiently?
- How do these depend on the interaction between f and G ?

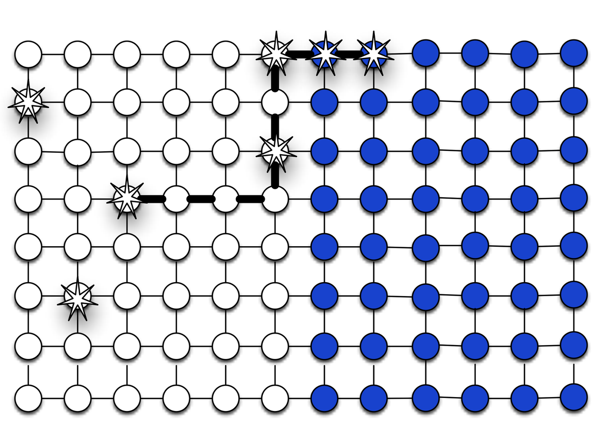
The S^2 Algorithm

1. **Randomly query** vertices till we find a pair of oppositely labeled vertices.
2. Ask for the label of the vertex at the midpoint of the shortest among all the shortest paths that connect oppositely labeled vertices. That is, **bisect the shortest shortest path** connecting oppositely labeled vertices.
   a. If two oppositely labeled vertices are connected, remove this **cut edge**.
3. Repeat **Step 2** till **no oppositely labeled pairs are connected.**
4. Return to **Step 1.**

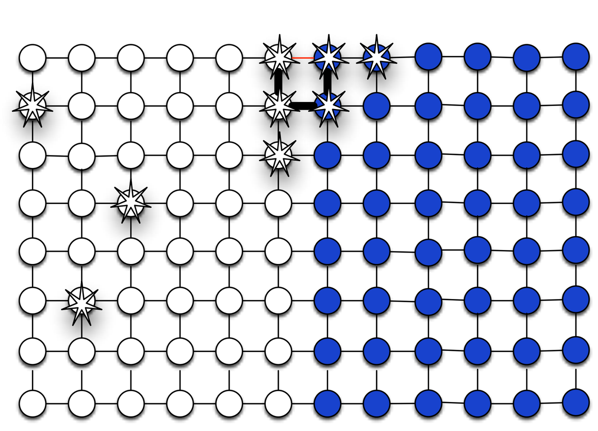
Example



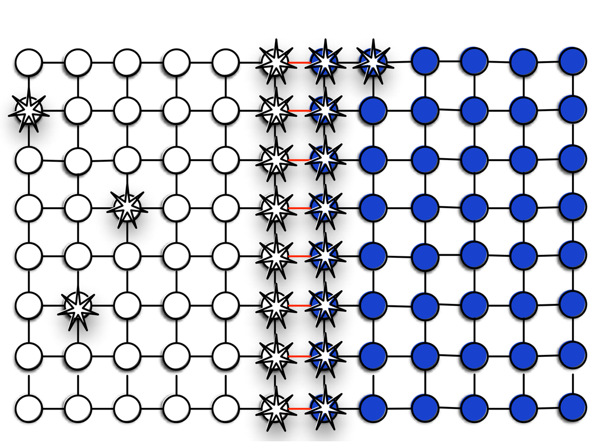
1. Random sampling ends. (One) Shortest Shortest path is shown



2. This path is bisected (repeatedly). S^2 finds one cut edge.



3. Next shortest shortest path is bisected.



4. This continues until S^2 **traces the cut boundary.**

arXiv:1506.08760

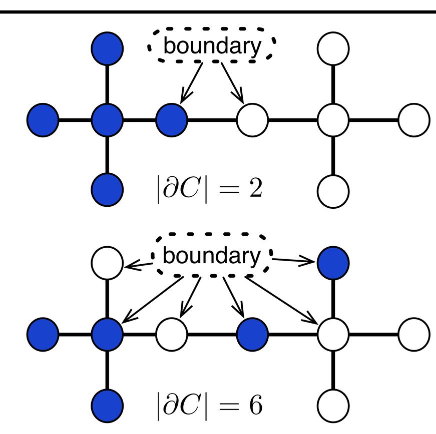
Measures of Complexity

The problem of predicting all the values of f from a subset $L \subset [n]$ is ill-posed.
If f can arbitrarily be any of the 2^n binary assignments on [n], $f(L)$ has no predictive power about $f([n] \setminus L)$

Boundary size $(|\partial C|)$

Number of vertices adjoining cut edges.

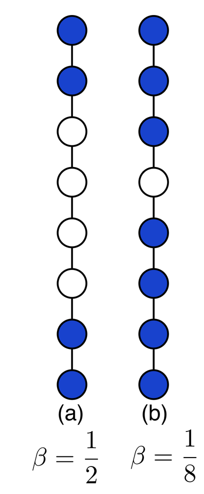
The smaller the better



Balancedness (β)

Ratio of the size of the smallest monochromatic component to n.

The bigger the better

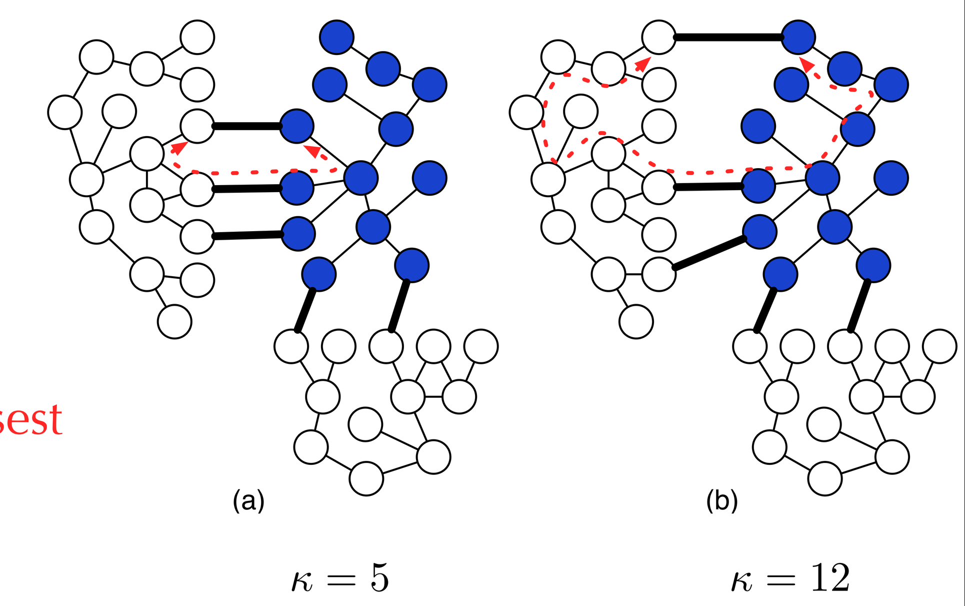


Clusteredness of the Cut Set (κ)

The cut set is composed of subsets that connect the same monochromatic components of the graph.

The clusteredness of each such subset is roughly the maximum (shortest path) distance between a cut edge and its closest cut edge in that subset.
The clusteredness of the cut set is the maximum of this quantity over all the subsets.

Given a cut edge, this novel complexity measure quantifies the ease of finding a new cut edge.



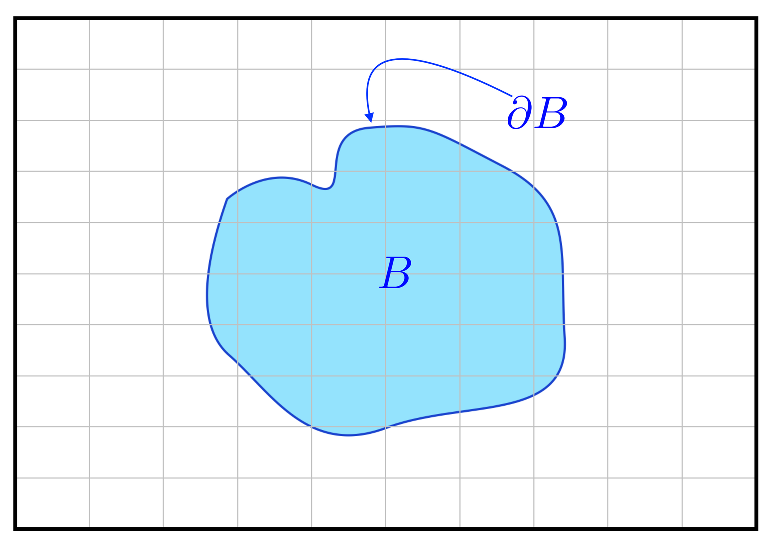
The smaller the better

Theorem : Suppose graph G and binary function f are such that they induce a cut set C with m components that are κ-clustered and such that the boundary size and balancedness are respectively $|\partial C|$ and β .
The #queries S2 needs to learn C is (roughly) bounded from above by

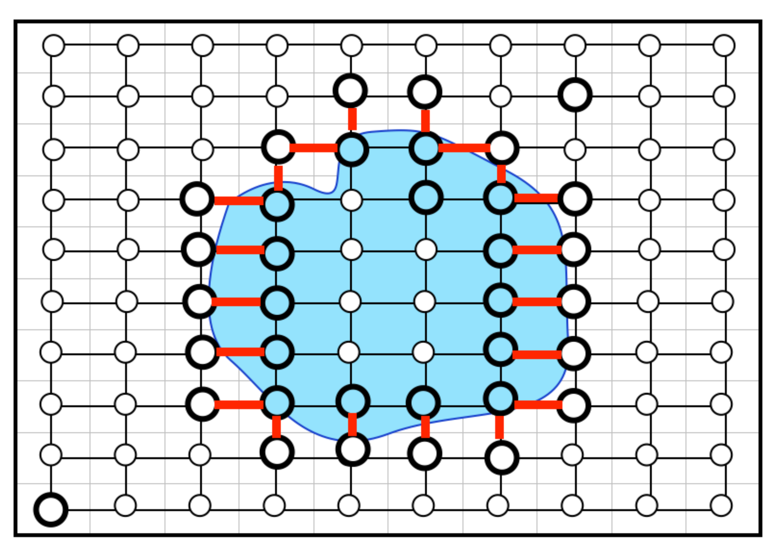
$$\log\left(\frac{1/\beta\epsilon}{1/(1-\beta)}\right) + m \log n + |\partial C| \log \kappa$$

ϵ - desired prob. of error

S^2 for Nonparametric Active Learning



$B \subset [0,1]^d$ represents (one class of the) classification rule. For any integer w, consider a partition of $[0,1]^d$ into cells of side length $1/w$
Assumption: #cells intersected by boundary ∂B is no more than cw^{d-1} (roughly, boundary is $d-1$dim.)
Called the *box-counting dimension*.



S^2 efficiently traces the boundary and learns B

Create a lattice graph on $[0,1]^d$ by assigning a vertex to each cell above; connect vertices corresponding to neighboring cells.

Run S^2 on this graph.

In this context, by querying a vertex, we mean picking $\mathcal{O}(\log w^d)$ points uniformly at random from the corresponding cell and taking the majority vote of the returned labels.

Theorem : Consider a classification problem whose Bayes optimal decision boundary has a box-counting dimension of *d-1*. If one runs S^2 as above with n samples, there is a constant C such that the excess risk is no more than $C\,(\log n/n)^{1/d-1}$ for n large enough (near minimax optimal).