Resistive geometry for graph-based transduction
EMMDS 2009

Mark Herbster

University College London
Department of Computer Science
Centre for Computational Statistics and Machine Learning

4 July, 2009

with Guy Lever, Massimiliano Pontil, and Sergio Rojas-Galeano

Outline

1. Introduction: transduction with a resistive network

2. Fast prediction on a tree

3. Generalized $p$-resistive networks
Part I
Transduction with a resistive network

Supervised learning

- Typical goal
  - Given data (pattern,target) \( S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\} \)
    infer \( f \) such that \( f(x_i) \approx y_i \) for future data
    \( S' = \{(x_{\ell+1}, y_{\ell+1}), (x_{\ell+2}, y_{\ell+2}), \ldots\} \).

- Algorithms
  1. Linear Regression
  2. Neural Networks
  3. Decision Trees
  4. Support Vector Machines
Unsupervised learning

- Typical goal
  - Given data (patterns) $S = \{x_1, \ldots, x_n\}$
    
    Model the data.

  - For example (clustering)
    
    Give a partition/function of $f : x \rightarrow \{1, \ldots, k\}$ of $S$

    Such that
    
    $f(x_i) = f(x_j) \Rightarrow x_i \approx x_j$

    $f(x_i) \neq f(x_j) \Rightarrow x_i \not\approx x_j$

- Algorithms
  1. Clustering (k-means)
  2. Dimensionality Reduction (PCA)

Semi-supervised Learning (Transductive case)

- Typical goal
  - Given data $S = \{(x_1, y_1), \ldots, (x_\ell, y_\ell), x_{\ell+n}, \ldots, x_n\}$
    
    infer $f$ such that $f(x_i) \approx y_i$ (No future data)

- Idea
  - Combine supervised and unsupervised learning.
  - Model data as a graph.

- Graph representation of data
  - Intrinsic: (protein interaction network, social network)
  - Built via a “distance” $d(p, q)$
    
    [sparse unweighted]: $k$-nearest neighbors graph
    
    [complete weighted]: Weight edge $(p, q)$ with $w_{pq} = \frac{1}{d(p, q)}$
Graph Examples – 1

Yeast protein network  Internet hosts

Graph Examples – 2

Web Spam  Twitter Social Network
Graph Examples – 3

USPS digits 3 and 8

Some questions

1. How to label the unknown vertices?
2. How to efficiently compute the labels of the unknown vertices?
3. How to guarantee the learning quality of the labelling?
4. How to build the graph?
Graph to resistive network

- Identify graph with a resistive network
- Labels are voltage constraints.
- Power of a labeling

$$P(u) := \|u\|_{G,2}^2 = \sum_{(i,j) \in E(G)} w_{ij} |u_i - u_j|^2$$

- Label by minimizing power

$$\bar{u} = \arg \min_{u \in \mathbb{R}^n} \{ P(u) : u_i = y_1, \ldots, u_i = y_\ell \}$$

- Predict with sign($\bar{u}$)

Effective resistance

Graph Laplacian $G := D - W$

- $W$ is the weighted adjacency matrix
- $D := \text{diag}(\sum_{i=1}^{n} W_{i1}, \ldots, \sum_{i=1}^{n} W_{in})$ (vertex degree matrix)
- $G^+$ is the “kernel” (pseudoinverse of $G$)

Theorem[KR93]

Effective resistance between $v_p$ and $v_q$ is

$$r_G(p, q) = \left[ \min_{u \in \mathbb{R}^n} \{ P(u) : u_p = 1, u_q = 0 \} \right]^{-1} = G_{pp}^+ + G_{qq}^+ - 2G_{pq}^+$$

Graph is a circuit and edge $(i, j)$ is a resistor $\pi_{ij} := w_{ij}^{-1} = d(i, j)$

- Effective resistance is 1 from $v_p$ to $v_q$
The “Intuition”
From “distance” to “resistance”
The base global “distance” \( d(p, q) \) is adapted according to the empirical distances via the effective resistance \( r(p, q) \).

Two \( m - \)cliques connected by an edge with distances \( a < b \)

- What is the label at “??”
- Effective resistance between vertices \( r_{m-\text{clique}} = \frac{2d}{m} \)
- If \( \frac{2b}{m} < a \) then label “+1”
- Conclusion: labelings respect cluster structure

Trees

Part II
Fast prediction on a tree
Overview

- Aim: Speed up Laplacian-based semi-supervised learning
- Model: large graph with small label and test set
- Method:
  1. Approximate graph with a tree
  2. Compute Lap. kernel quickly via resistive network analogy
- Experiments:
  1. Approximate with MST and SPT trees (ensembles)
  2. Predict with kernel perceptron
  3. Web graph data 400,000 vertices 10 million+ edges

Setup

- Predict only a small subset of points
- Expectation: \( l + p \ll u \)

Legend:

- \( l \): # labeled points
- \( p \): # test points
- \( u \): # unlabeled points
Graph connectivity

Inverse Connectivities

Graph: \( R_{\text{tot}} := \sum_{i>j} r_G(i,j) \)

Vertex:
1. \( R(v) := \sum_{i=1}^{n} r_G(v,i) \)
2. \( G_{vv}^+ = \frac{R(v)}{n} - \frac{R_{\text{tot}}}{n^2} \)

Grey-scale \( G_{vv}^+ : G_{30,30}^+ = .21 \) (min), \( G_{15,15}^+ = .94 \) (max)

Effective resistance on a tree

Effective resistance from \( v_3 \) to \( v_4 \)

\[ r_T(3, 4) = \pi_{13} + \pi_{12} + \pi_{24} \]

“Resistors in series” : sum along unique path
Computing the diagonal in $O(n)$ time

1. Compute $R(v)$, $v = 1, \ldots, n$. 
2. $R_{\text{tot}} = \frac{1}{2} \sum_{v=1}^{n} R(v)$. 
3. $G_{v,v}^+ = \frac{R(v)}{n} - \frac{R_{\text{tot}}}{n^2}$. 

Computing an $R(v)$

1. Define $T(v) = \sum_{x \in \text{descendants}(v)} r_G(v, x)$. 
2. Define $T'(v) = \sum_{x \not\in \text{descendants}(v)} r_G(v, x)$. 
3. $R(v) = T(v) + T'(v)$. 

Computing $m \times m$ block of the kernel $G^+$ (tree)

Computing the off-diagonal

Recall: $G_{i,j}^+ = \frac{1}{2} (G_{i,i}^+ + G_{j,j}^+ - r_G(i,j))$

1. Single: $G_{i,j}^+$ in $O(S)$ ($S$ is diameter) 
2. Amortized Block ($m \times m$): $O(m^2 + mS)$

Amortized algorithm for $m \times m$ block

1. Input: $\{v_1, \ldots, v_m\} \subseteq V$
2. Initialization: $\text{visited}(\text{all}) = \emptyset$
3. for $i = 1, \ldots, m$ do
4.   $p = -1; c = v_i; r_T(c, c) = 0$
5.   Repeat
6.     for $w \in \text{visited}(c) \cap \{p\} \cup \uparrow(p)$ do
7.       $r_T(v_i, w) = r_T(w, v_i) = r_T(v_i, c) + r_T(c, w)$
8.     end
9.   $\text{visited}(c) = \text{visited}(c) \cup v_i$
10. $p = c; c = \uparrow(c)$
11. $r_T(v_i, c) = r_T(c, v_i) = r_T(v_i, p) + \pi_p, c$
12. until (“$p$ is the root”) 
13. end

- “Initialization” + amortized algorithm: $O(n + m^2 + mS)$
Approximate the graph

- What if the graph is not a tree?
- Approximate with minimum or shortest paths spanning tree
- Use ensembles of trees

### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Agg. (81/21)</th>
<th>AUC</th>
<th>Single</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host-graph</strong> (9,072 vertices, 464,959 edges; aggregates: 81)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MST</td>
<td>0.907</td>
<td>0.950</td>
<td>0.857±0.022</td>
<td>0.841±0.045</td>
</tr>
<tr>
<td>SPT</td>
<td>0.889</td>
<td>0.952</td>
<td>0.850±0.026</td>
<td>0.804±0.063</td>
</tr>
<tr>
<td>MST (bidir)</td>
<td>0.912</td>
<td>0.944</td>
<td>0.878±0.033</td>
<td>0.851±0.100</td>
</tr>
<tr>
<td>SPT (bidir)</td>
<td><strong>0.913</strong></td>
<td><strong>0.960</strong></td>
<td>0.873±0.028</td>
<td>0.846±0.065</td>
</tr>
<tr>
<td>Abernathy et al.</td>
<td>0.896</td>
<td>0.952</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Tang et al.</td>
<td>0.906</td>
<td>0.951</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Filoche et al.</td>
<td>0.889</td>
<td>0.927</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Benczúr et al.</td>
<td>0.829</td>
<td>0.877</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td><strong>Web-graph</strong> (400,000 vertices, 10,455,544 edges; aggregates: 21)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MST (bidir)</td>
<td>0.991</td>
<td><strong>1.000</strong></td>
<td>0.976±0.011</td>
<td>0.993±0.005</td>
</tr>
<tr>
<td>SPT (bidir)</td>
<td>0.994</td>
<td>0.999</td>
<td>0.985±0.002</td>
<td>0.992±0.003</td>
</tr>
<tr>
<td>Witschel et al.</td>
<td><strong>0.995</strong></td>
<td>0.998</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Filoche et al.</td>
<td>0.973</td>
<td>0.991</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Benczúr et al.</td>
<td>0.942</td>
<td>0.973</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Tang et al.</td>
<td>0.296</td>
<td>0.989</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Classifier: kernel perceptron
- “bidir”: double-weight mutually linked edges
Aggregate performances

Accuracy versus number of trees

$p$-Resistance

Part III
Generalized $p$-resistive networks
**p-Resistive Networks**

- Inspired by the p-norm perceptron [GLS97]
- Redefine power as
  \[ P_p(u) := \|u\|_{G,p}^p = \sum_{(i,j) \in E(G)} w_{ij} |u_i - u_j|^p \]

- Analogues of usual “electric network” theory now follow ...
  1. Kirchoff’s laws
  2. Ohm’s law
  3. Conservation of energy principle
  4. Rayleigh’s monotonicity principle
  5. Black box principle (2-port)
  6. The “rules” of resistors in parallel and in series,

---

**p-Resistance Definition**

The (effective) p-resistance between any two vertices \(a\) and \(b\) is

\[
r_p(a, b) = \left( \min_{u \in \mathbb{R}^n} \{ P_p(u) : u_a = 1, u_b = 0 \} \right)^{-1}.
\]

**Resistors in series**

\[
r_p(a, b) = \left( \sum_{i=1}^{n} \pi_i^{p-1} \right)^{p-1}
\]

**Resistors in parallel**

\[
r_p(a, b) = \left( \sum_{i=1}^{\pi_1} \frac{1}{\pi_i} \right)^{-1}
\]

Note if \(p = 1\) then \(r_{G,1}(a, b) = \frac{1}{\text{mincut between } a \text{ and } b}\)
Online Learning Model

- Aim: learn a function \( u : V \to \{-1, +1\} \) corresponding to a labeling of a graph \( G = (V, E) \) and \( V = \{1, \ldots, n\} \).

- Learning proceeds in trials

\[
\text{for } t = 1, \ldots, \ell \text{ do}
\]

1. Nature selects \( v_t \in V \)
2. Learner predicts \( \hat{y}_t \in \{-1, +1\} \)
3. Nature selects \( y_t \in \{-1, +1\} \)
4. If \( \hat{y}_t \neq y_t \) then mistakes = mistakes + 1

- Learner’s goal: minimize mistakes
- mistakes \( \leq f(\text{complexity}(u), \text{structure}(G)) \)

Power interpolation algorithm

- Choose parameter \( 1 < p \leq 2 \)
- Given a sequence of online trials

\[
\{(v_{i_1}, y_1), (v_{i_2}, y_2), \ldots (v_{i_\ell}, y_\ell)\}
\]

- Hypothesis vector is interpolant

\[
\tilde{u}_t = \arg \min_{u \in \mathbb{R}^n} \{ P_p(u) : u_{i_1} = y_1, \ldots, u_{i_\ell} = y_\ell \}
\]

- Predict

\[
\hat{y}_t = \text{sign}(\tilde{u}_{i_t})
\]

- As \( p \to 1 \) predictions correspond to label-consistent mincut
Mistake Bound

Theorem
Given $p \in (1, 2]$ then

$$M \leq N_\rho + \frac{[\rho \times P_p(u)]^2}{p - 1}$$

for all $0 < \rho$, and for all consistent $u \in \mathbb{R}^n$.

- Relative to any resistive cover w.r.t. $r_p$
- Both number of covering sets $N_\rho$ and resistance $\rho$

Example: Octopus

A $d$-tentacle octopus graph with length $d$ tentacles ($n = d^2 + 1$)

- Cover: $N = 1$, Power: $P_p(u) = 1 \times 2^p$
- Connectivity: $k = 1$, Diameter: $\Delta_1 = 2d$
- When $p = 1 + \frac{1}{\log(2d)-1}$ then $M_A \leq 1 + 4e^2(\log(2d) - 1)$
- When $p = 2$ then $\sqrt{n} \leq M_A \leq O(\sqrt{n})$
Example: Prototypical Clusters

\[
c \times m\text{-cliques with } \ell \text{ cut edges}
\]

- Cover: \( N = c \), Power: \( P_p(u) = \ell \times 2^p \)
- Connectivity: \( k = m - 1 \), Wide Diameter: \( \Delta_{m-1} = 2 \)
- When \( p = 2 \) then \( M \leq c + \frac{8\ell}{m-1} \)

Conclusions

- Graph labelings predicted via power minimization
- Base distances were transformed to effective resistances
- Eff. resistance demonstrated to respect cluster structure
- Fast computation of tree kernels via effective resistance
- A \( p \)-resistive network theory may be given
- Selection of \( p \) enables logarithmic mistake bounds
Thanks